Locality in space and time for data-efficient visual recognition

Kayhan, O.S.

DOI
10.4233/uuid:983fb3e9-2aa8-4161-a697-3c36e0dcbcbb

Publication date
2022

Document Version
Final published version

Citation (APA)
Kayhan, O. S. (2022). *Locality in space and time for data-efficient visual recognition*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:983fb3e9-2aa8-4161-a697-3c36e0dcbcbb

LOCALITY IN SPACE AND TIME FOR DATA-EFFICIENT VISUAL RECOGNITION

Osman Semih Kayhan

LOCALITY IN SPACE AND TIME FOR DATA-EFFICIENT VISUAL RECOGNITION

Propositions

accompanying the dissertation

LOCALITY IN SPACE AND TIME FOR DATA-EFFICIENT VISUAL RECOGNITION

by

Osman Semih KAYHAN

1. CNNs excel at exploiting biases in datasets. (this thesis)
2. Data augmentation cannot satisfy the data hunger of deep networks. (this thesis)
3. Convolution was, is, and will be a cornerstone of computer vision.
4. Accusing researchers of creating ethically biased machine learning algorithms resembles a modern day witch-hunt.
5. There is no unfair algorithm, there is always an unfair human being.
6. The Internet is evolving into the biggest platform for controlled experiments.
7. There is no catastrophic forgetting, every decision in the past affects the future.
8. Positive discrimination is not a solution for diversity.
9. Abundance will bring poverty.
10. Mixing cultures brings high development.

These propositions are regarded as opposable and defendable, and have been approved as such by the promotor Prof. dr. ir. M. J. T. Reinders.

LOCALITY IN SPACE AND TIME FOR DATA-EFFICIENT VISUAL RECOGNITION

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board of Doctorates,
to be defended publicly on
Monday 21 February 2022 at 15:00 o'clock

by

Osman Semih KAYHAN

Master of Science in Biomedical Engineering,
Istanbul Technical University, Turkey
Born in Afyonkarahisar, Turkey

This dissertation has been approved by the

promotor: Prof. dr. ir. M. J. T. Reinders
copromotor: Dr. J. C. van Gemert

Composition of the doctoral committee:

Rector Magnificus,	Chairperson
Prof. dr. ir. M. J. T. Reinders,	Delft University of Technology, promotor
Dr. J. C. van Gemert,	Delft University of Technology, copromotor
Independent members:	
Prof. dr. A. Hanjalic	Delft University of Technology
Prof. dr. S. C. Pont	Delft University of Technology
Prof. dr. C. G. M. Snoek	University of Amsterdam
Prof. dr. A. A. Salah	Utrecht University
Dr. H. Dibeklioğlu	Bilkent University
Dr. H. Hung	Delft University of Technology, reserve member

Printed by: ProefschriftMaken.

Cover photo: Çağrı Selek.

Cover design: Oğuzhan Çilenk.

Karahisar Castle is a historical fortification which was built around 1350 BC by The Hittite king Mursilis II. Over years, the city has been located around the Castle. The Castle has always been a location indicator since its existence. It also indicates the era of all the civilizations and carries traces of them. Namely, Karahisar Castle denotes locality in space and time. The city is named Afyonkarahisar as the Castle is the symbol of it.

ISBN 978-94-6384-302-7

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

To my family

CONTENTS

SUMMARY

Spatial localization in time is vital for humans. Therefore we desire that computer vision algorithms are also able to spatially and temporally localize objects and actions. These algorithms generally learn from given data and discover patterns, parts, motions, and their locations by exploiting inductive biases that are essential for learning. However, localization is complex, error-prone and hard to inspect. In this thesis, we investigate location biases and how CNNs explore and exploit location and temporal information in the image and video domain.

An interesting finding of the thesis is that heuristics about what is outside the image (border handling) enables CNNs to exploit absolute spatial location and break translation equivariance. The thesis proposes a simple solution to eliminate the spatial location biases. The proposed solution improves translation equivariance and provides data efficiency and robustness.

Furthermore, the thesis investigates object and part locations on images. First, the thesis studies object-context relationships of modern object detectors and reveals insights about helpful location biases. In addition, the effect of unhelpful location biases is investigated for a visual verification task. These analyses show that object detectors can hallucinate the location of an object with high confidence score even if the object is not in the image. Based on these insights, the thesis provides suggestions for researchers on how to choose an object detector for their specific tasks.

Another interesting finding of this thesis shows limitations of data augmentation techniques to resolve robustness issues of pose estimation methods when dealing with occlusions. Even if data augmentation alleviates some problems caused by sampling biases, it can only yield limited improvement and the performance saturates after applying a stack of augmentations.

Finally, the thesis investigates temporal location information and demonstrates spatio-temporal location biases in video data. A time-efficient video labeling solution that uses latent space feature similarity is proposed to annotate long-untrimmed videos. Besides, using only keyframe labels with Positive-Unlabeled learning achieves high-quality action proposals that can be utilized with many temporal action localization methods. The proposed method can provide data and label efficiency.

Taken together, this thesis investigates how CNNs use location information and introduce location biases that can result in positive as well as negative outcomes on various computer vision tasks.

SAMENVATTING

Ruimtelijke lokalisatie in de tijd is van belangrijk voor de mens. Daarom willen we dat computer vision-algoritmen ook in staat zijn om objecten en acties ruimtelijk en temporeel te lokaliseren. Deze algoritmen leren uit data en ontdekken patronen, onderdelen, bewegingen en hun locaties door gebruik te maken van inductieve vooroordelen die essentieel zijn voor leren. Lokalisatie is echter complex, foutgevoelig en moeilijk te inspecteren. In dit proefschrift onderzoeken we locatievooroordelen en hoe CNN's locatie- en temporele informatie in het beeld- en videodomein verkennen en exploiteren.

Een interessante bevinding van het proefschrift is dat heuristieken over wat zich buiten het beeld bevindt de CNN's in staat stelt om de absolute ruimtelijke locatie te benutten en de positie-equivariantie te doorbreken. Het proefschrift stelt een eenvoudige oplossing voor om de ruimtelijke locatiebias te elimineren. De voorgestelde oplossing verbetert de positie-equivariantie en zorgt voor data-efficiëntie en robuustheid.

Verder onderzoekt het proefschrift object- en onderdeel-locaties in afbeeldingen. Ten eerste bestudeert het proefschrift object-context relaties van moderne objectdetectoren en onthult inzichten over nuttige locatievooroordelen. Daarnaast wordt het effect van locatie vooroordelen onderzocht voor een visuele verificatietaak. Deze analyses tonen aan dat objectdetectoren de locatie van een object met een hoge betrouwbaarheidsscore kunnen hallucineren, zelfs als het object niet in het beeld staat. Op basis van deze inzichten biedt het proefschrift onderzoekers suggesties voor het kiezen van een objectdetector voor hun specifieke taken.

Een andere interessante bevinding van dit proefschrift toont beperkingen aan van data-augmentatietechnieken om robuustheidsproblemen van pose - schattingsmethoden op te lossen bij het omgaan met occlusies. Zelfs als gegevensvergroting enkele problemen verlicht die worden veroorzaakt door bemonsteringsbias, kan dit slechts een beperkte verbetering opleveren en de prestaties verzadigen na het toepassen van veel augmentaties.

Ten slotte onderzoekt het proefschrift tijdelijke locatie-informatie en demonstreert het ruimtelijk-temporele locatievooroordelen in videogegevens. Er wordt een tijdbesparende oplossing voor het labelen van video's voorgesteld die gebruikmaakt van gelijkenis van latente ruimtefuncties om lange ongeknipte video's te annoteren. Bovendien levert het gebruik van alleen keyframe-labels met Positive-Unlabeled learning actievoorstellen van hoge kwaliteit op die kunnen worden gebruikt met veel lokalisatiemethoden voor tijdelijke acties. De voorgestelde methode kan gegevens- en labelefficiëntie opleveren.

Alles bij elkaar genomen, onderzoekt dit proefschrift hoe CNN's locatie-informatie gebruiken en locatievooroordelen introduceren die zowel positieve als negatieve resultaten kunnen opleveren bij verschillende computervisietaken.

1

INTRODUCTION

1

Well before Einstein physically linked space and time, there was already a strong semantic –humanly meaningful– connection between the *here* and the *now*. The time of sunset is linked to a safe place to sleep; while sunrise unlocks hunting grounds and berry locations. Mapping such important locations describes the known world, where beyond the boundaries of such maps, the unknown lurks. Going beyond the boundaries and exploring these unknown locations led to several important historical events in time such as the circumnavigation of Africa, and linking Europe to the Americas. In modern times, where social media and the internet hosts our digital images and videos, the connection between time and space is exemplified by photos on a timeline, images and videos with GPS tags, automatically generated moving image slideshows, etc. Because humans derive meaning from time and space, the *where* and the *when* permeate our digital images and videos.

Automatic analysis tasks for images and videos reflect the importance of time and space. Where an automatic image classification task has a computer assign a label to the whole image, an object detection task also requires a precisely localized box around the detected objects. For detecting small parts, such as a wrist, shoulder, knee, etc, as used in automatic human pose estimation, the part location is given by predicting a point per part. Similarly, for automatic video analysis, the absolute position in time plays no role in action recognition, as an action may occur anywhere in the video. Yet, for temporal action localization, the start and end times of an action are also required. How spatial and temporal positions are handled, determines what automatic image and video analysis task are relevant for the application at hand.

The current approaches to automatic image and video analysis are based on deep learning. Deep learning is a type of machine learning based on neural networks, where instead of giving precise instructions on how to do a task, the machine is given many examples of what the outcome of a task should be. Thus, when spatial or temporal location information is required in the automatic analysis, a deep learning system needs examples that have the relevant location information annotated. Such annotations are expensive, as they require human effort, where the amount of effort is correlated with the precision of the annotated location information: annotating if an object is present in an image is less effort than annotating its location by a bounding box, which in turn, is less effort than annotating precise part locations. Annotating spatial-temporal locations in videos adds an additional temporal dimension which increases annotation efforts further. Deep learning approaches are crucially dependent on large amounts of annotated examples, and there is substantial human effort required in annotating these examples.

It is impossible to collect and annotate all possible locations for an object in images and videos. The object position in an image depends on the arbitrary camera location and viewpoint and can vary arbitrarily. Thus, machine learning methods are trained on a subset of all possible locations. This subset of locations, in turn, then determines what the machine learns, and it is thus not guaranteed that a machine learning approach trained on one particular subset of locations will generalize to recognizing a different subset of locations. For example, when humans take a photograph, the sky is typically in the top of the image. In contrast, a camera mounted on a flying drone does not have this bias, and during its maneuvers might even fly upside down, yielding the sky at the bottom of the image. A machine learning algorithm trained on object locations in pho-

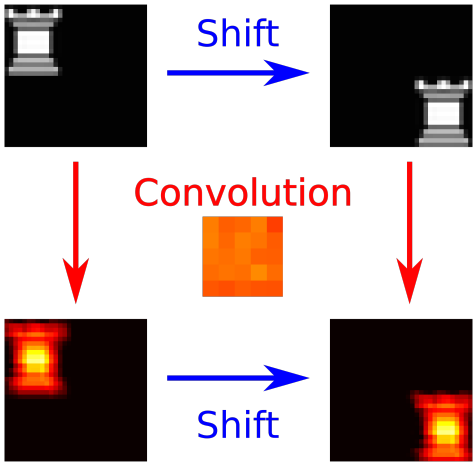


Figure 1.1: Example of convolution. The top row illustrates a shift of the input image, where the rook is moved from the top left to the bottom right of the image. The response of applying the convolution operator with the indicated filter in the center, is illustrated in the bottom row: If the patch shifts, the response shifts accordingly from the top left to the bottom corner as well.

tographs taken by humans, might thus not recognize locations in drone images.

Instead of a data-driven approach, which relies on representative data samples of all relevant object locations, a powerful alternative is to remove the ability to exploit location information from the machine learning algorithm. Incorporating such prior knowledge can thus substantially reduce the data collection and annotation effort. For image recognition approaches, the current default approach is to use a CNN (Convolutional Neural Network) which has exactly this goal: A CNN aims to remove location information by adding the convolution operator to a deep network. A convolution can be seen as forcing the machine learning algorithm to use a sliding window over *all* locations in the image or video, which should make it impossible to single out specific locations. To give an example, in Fig. 1.1, if a patch in an input image is shifted, the outcome of a convolution is equivalently shifted. The CNN is the current dominant deep learning image recognition paradigm, illustrating the role of location —or to be more precise, the power of ignoring location— in current visual machine learning approaches.

Humans care about location information in images and videos and consequently, location plays an important role in automatic visual analysis applications. Automating such application is well suited to a machine learning approach, and deep learning in particular. Machine learning methods depend crucially on giving examples and thus rely on valuable human annotations. The annotation effort can severely be reduced by adding prior knowledge to machine learning, where the marriage of the convolution operator and deep learning yields the CNN, which is the current default image recognition approach. This PhD thesis revolves around these topics; it investigates location information in deep learning approaches for images and videos while reducing the required annotation-effort.

Topic	Thesis chapter 2	3	4	5	6	7
Position in images						
Object classification	X					
Object detection		X	X			
Part detection				X		
Position in videos						
Action recognition	X				X	
Action localization						X
Position bias						
Absolute	X					
Relative		X	X	X		
Annotation reduction						
Efficient labeling					X	
Data efficiency	X					X

Table 1.1: Topic distribution over the thesis chapters. The thesis studies locality in image and videos while reducing the annotation effort.

1.1. FOCUS OF THE THESIS

The distribution of topics over the thesis chapters is given in Table 1.1.

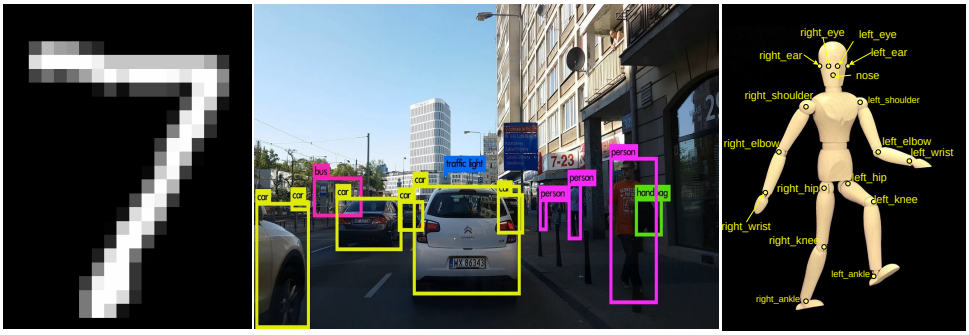


Figure 1.2: (left) Handwritten digit classification [1], (middle) the detection of the objects on the road [2], (right) Human body part locations [3].

Position in images. The considered automatic applications using location for images include *object classification:* deciding if an object is present somewhere in the image; *object detection:* predicting object location in the form of a rectangular bounding box around the object; *part detection:* predicting human part locations such as wrist, shoulder, head, etc., as a 2D point. Fig. 1.2 illustrates the three types of position considered in the thesis.

Position in videos. The considered location applications for video are *action recog-*

nition: deciding if an action occurs in any time in the video, and *action localization:* which also requires determining the temporal beginning and end of each action. Fig. 1.3 illustrates these two types of video positions considered in the thesis.

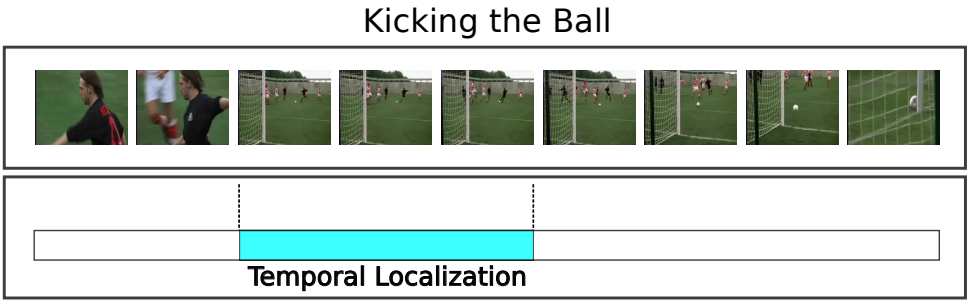


Figure 1.3: Visualization of *Kicking the Ball* action video [4]. Action recognition task only indicates the action class. For action localization task, in addition to the action class, temporal bounds of the action is required.

Position bias. The thesis investigates two types of position bias; *Absolute position* can be indexed on a coordinate system, such as the (x, y) pixel positions and the frame number in a video. *Relative position* is about the relative positional relation between the object and its surroundings, such as the relation of a boat surrounded by water, or a human head above a torso. Fig. 1.4 illustrates these two types of position bias considered in the thesis.

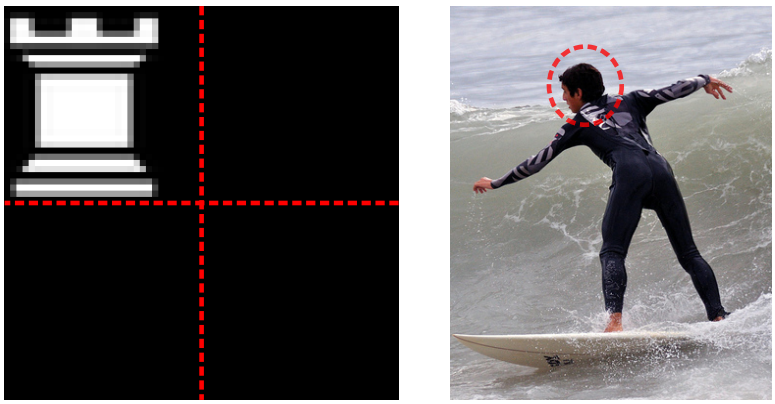


Figure 1.4: (left) Absolute position of rook [5], (right) Relative position of the surfer's head above his torso [6].

Annotation reduction. For reducing the annotation effort, the thesis considers *efficient labeling* which makes the labeling process itself faster and easier, while it also considers *data efficiency* which focuses on reducing the dependency of the deep learning algorithms on the amount of annotated data samples. Fig. 1.5a illustrates an example of making video labeling more efficient; whereas Fig. 1.5b shows the effect of data efficiency by losing less accuracy in the small data regime.

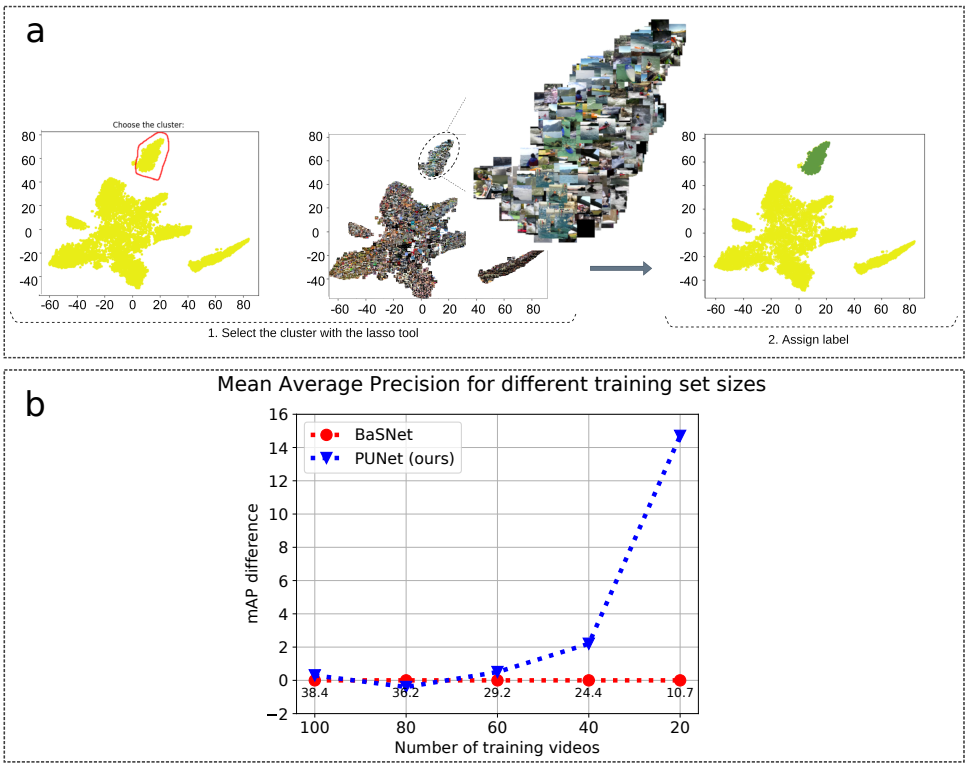


Figure 1.5: Annotation reduction. (a) Using group labeling makes annotation of videos more efficient [7]. (b) Using more data efficient methods provide comparable result with less labeled data [8].

1.2. OVERVIEW OF SUBSEQUENT CHAPTERS

The rest of the thesis is structured as follows. Chapter 2 investigates the convolution operator which powers the convolutional neural network (CNN) which is arguably the current strongest inductive prior in visual deep learning. The convolution can be seen as a sliding dot product and shares parameters at each location in the image, which attempts to remove the effect of absolute position information. The chapter shows that border effects, *i.e.* how the convolution operator is defined on the border of the image, allow common CNN architectures to exploit absolute position information in images and in video. By properly handling these border effects, the absolute position information is lost, offering increased data efficiency.

Chapter 3 evaluates the contextual relative object position on current deep learned object detectors. We qualitatively evaluate the effect of the object context on the three most prevalent types of object detectors and analyze correlations object/context correlations.

Chapter 4 explores object detection in images for automatic visual inspection. We define visual verification as determining if relevant object parts are present and at it's correct location. To evaluate visual verification, we introduce a new dataset of 10k im-

ages, where each image contains a bicycle. Each bicycle image has 22 densely annotated bike parts, where each part state is labeled as intact, missing, occluded or broken. We analyze how object detectors perform on this task, that common evaluation measures are not sufficient to evaluate visual verification, and show that object detectors are sensitive to part positions and thus might hallucinate non-existing parts as being present in the expected location.

Chapter 5 studies the role of relative locations for handling occlusions in body parts localization for human pose estimation using deep learning. A standard approach in deep learning is to use data augmentation which adds transformed copies of the input data to the deep network training set. Examples of data augmentation include different crops, locations, orientations, noise levels of input images. The chapter investigates how sensitive human pose estimation methods are to occlusions and how well data augmentation can pose a solution.

Chapter 6 explores how to efficiently annotate videos with spatio-temporal labels. To speed up the annotation effort, each video frame is mapped to a 2D point by using t-SNE, and by exploiting the redundancy between video frames, similar frames are mapped close to each other. This allows efficient annotation by allowing easy frame grouping in 2D, making it possible to annotate several frames at once.

Chapter 7 investigates data efficiency in temporal video action localization, where the goal is to segment actions from long, untrimmed videos. Data efficiency is achieved by a weakly-supervised action proposal network, *PUNet* which uses a single frame temporal label rather than temporal action bounds. A single frame is faster to annotate since the exact temporal bounds no longer need to be labeled.

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition,* Proceedings of the IEEE **86**, 2278 (1998).

[2] *Yolo v3 object detection with ros,* https://neilnie.com/2018/11/18/, accessed: 2021-09-05.

[3] R. Pytel, O. S. Kayhan, and J. C. van Gemert, *Tilting at windmills: Data augmentation for deep pose estimation does not help with occlusions,* in *2020 25th International Conference on Pattern Recognition (ICPR)* (2021) pp. 10568–10575.

[4] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, *Hmdb: a large video database for human motion recognition,* in *2011 International conference on computer vision* (IEEE, 2011) pp. 2556–2563.

[5] O. S. Kayhan and J. C. v. Gemert, *On translation invariance in cnns: Convolutional layers can exploit absolute spatial location,* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft coco: Common objects in context,* in *ECCV* (2014).

[7] S. Poorgholi, O. S. Kayhan, and J. C. v. Gemert, *t-eva: Time-efficient t-sne video annotation,* in *International Conference on Pattern Recognition* (Springer, 2021) pp. 153–169.

[8] N. U. S. Zia, O. S. Kayhan, and J. v. Gemert, *Punet: Temporal action proposal generation with positive unlabeled learning using key frame annotations,* in *2021 IEEE International Conference on Image Processing (ICIP)* (2021) pp. 2598–2602.

2

On Translation Invariance in CNNs: Convolutional Layers can Exploit Absolute Spatial Location

Science is the only true guide in life.

M. K. Atatürk

This chapter has been published as:
O. S. Kayhan and J. C. van Gemert, On translation invariance in cnns: Convolutional layers can exploit absolute spatial location, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) [1]

2

ABSTRACT

In this paper we challenge the common assumption that convolutional layers in modern CNNs are translation invariant. We show that CNNs can and will exploit the absolute spatial location by learning filters that respond exclusively to particular absolute locations by exploiting image boundary effects. Because modern CNNs filters have a huge receptive field, these boundary effects operate even far from the image boundary, allowing the network to exploit absolute spatial location all over the image. We give a simple solution to remove spatial location encoding which improves translation invariance and thus gives a stronger visual inductive bias which particularly benefits small data sets. We broadly demonstrate these benefits on several architectures and various applications such as image classification, patch matching, and two video classification datasets. [1]

[1] For the code:

https://github.com/oskyhn/CNNs-Without-Borders

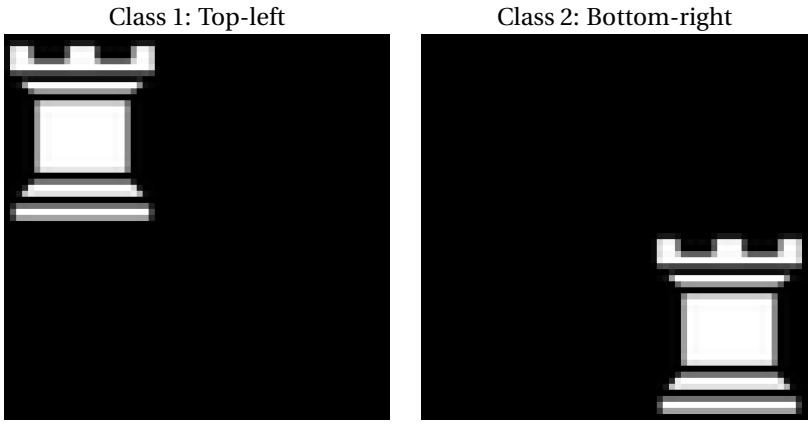


Figure 2.1: We place an identical image patch on the top-left or on the bottom-right of an image. We evaluate a standard fully convolutional network [3, 5, 33–36] if it can classify the patch location (top-left vs bottom-right). We use 1 layer, a single 5x5 kernel, zero-padding, same-convolution, ReLu, global max pooling, SGD, and a soft-max loss. Surprisingly, this network can classify perfectly, demonstrating that current convolutional layers can exploit the absolute spatial location in an image.

2.1. INTRODUCTION

The marriage of the convolution operator and deep learning yields the Convolutional Neural Network (CNN). The CNN arguably spawned the deep learning revolution with AlexNet [2] and convolutional layers are now the standard backbone for various Computer Vision domains such as image classification [3–5], object detection [6–8], semantic segmentation [9–11], matching [12–14], video [15–17], generative models [18–20], *etc.* The CNN is now even used in other modalities such as speech [21–23], audio [24–26], text [27–29], graphs [30–32], *etc.* It is difficult to overstate the importance of the convolution operator in deep learning. In this paper we analyze convolutional layers in CNNs which is broadly relevant for the entire deep learning research field.

For images, adding convolution to neural networks adds a visual inductive prior that objects can appear anywhere. Convolution can informally be described as the dot product between the input image and a small patch of learnable weights –the kernel– sliding over all image locations. This shares the weights over locations yielding a huge reduction in learnable parameters. Convolution is equivariant to translation: If an object is shifted in an image then the convolution outcome is shifted equally. When convolution is followed by an operator that does not depend on the position, such as taking the global average or global maximum, that gives translation invariance and absolute location is lost. Translation invariance powers the visual inductive prior of the convolution operator, and we will demonstrate that improving translation invariance improves the prior, leading to increased data efficiency in the small data setting.

In this paper we challenge standard assumptions about translation invariance and show that currently used convolutional layers can exploit the absolute location of an object in an image. Consider Fig. 2.1, where the exactly identical image patch is positioned on the top left (class 1) or on the bottom right (class 2) in an image. If a fully convolu-

tional CNN is invariant, it should not be able to classify and give random performance on this task. Yet, surprisingly, a simple standard 1-layer fully convolutional network with a global max pooling operator can perfectly classify the location of the patch and thus exploit absolute spatial location.

We show that CNNs can encode absolute spatial location by exploiting image boundary effects. These effects occur because images have finite support and convolving close to the boundary requires dealing with non-existing values beyond the image support [37, 38]. Boundary effects allow CNNs to learn filters whose output is placed outside the image conditioned on their absolute position in the image. This encodes position by only keeping filter outputs for specific absolute positions. It could, for example, learn filters that only fire for the top of the image, while the bottom responses are placed outside the image boundary. Boundary effects depend on the size of the convolution kernel and are small for a single 3x3 convolution. Yet, CNNs stack convolution layers, yielding receptive fields typically several times the input image size [39]. Boundary effects for such huge kernels are large and, as we will demonstrate, allows CNNs to exploit boundary effects all over the image, even far away from the image boundary.

We have the following contributions. We show how boundary effects in discrete convolutions allow for location specific filters. We demonstrate how convolutional layers in various current CNN architectures can and will exploit absolute spatial location, even far away from the image boundary. We investigate simple solutions that removes the possibility to encode spatial location which increases the visual inductive bias which is beneficial for smaller datasets. We demonstrate these benefits on multiple CNN architectures on several application domains including image classification, patch matching, and video classification.

2.2. RELATED WORK AND RELEVANCE

Fully connected and fully convolutional networks. Initial CNN variants have convolutional layers followed by fully connected layers. These fully connected layers can learn weights at each location in a feature map and thus can exploit absolute position. Variants of the seminal LeNet that included fully connected layers experimentally outperformed an exclusively convolutional setup [40]. The 2012 ImageNet breakthrough as heralded by AlexNet [2] followed the LeNet design, albeit at larger scale with 5 convolutional and 2 fully connected layers. Building upon AlexNet [2], the VGG [4] network family variants involve varying the depth of the convolutional layers followed by 3 fully connected layers. The fully connected layers, however, take up a huge part of the learnable parameters making such networks large and difficult to train.

Instead of using fully connected layers, recent work questions their value. The Network In Network [34] is a fully convolutional network and simply replaces fully connected layers by the global average value of the last convolutional layer's output. Such a global average or global max operator is invariant to location, and makes the whole network theoretically insensitive to absolute position by building on top of equivariant convolutional layers. Several modern networks are now using global average pooling. Popular and successful examples include the The All Convolutional Net [35], Residual networks [3], The Inception family [5], the DenseNet [33], the ResNext network [36] *etc.* In this paper we show, contrary to popular belief, that fully convolutional networks will

exploit the absolute position.

Cropping image regions. Encoding absolute location has effect on cropping. Examples of region cropping in CNNs include: The bounding box in object detection [8, 9, 41]; processing a huge resolution image in patches [42, 43]; local image region matching [12–14, 44]; local CNN patch pooling encoders [45–47]. The region cropping can be done *explicitly* before feeding the patch to a CNN as done in R-CNN [41], high-res image processing [42] and aggregation methods [48, 49]. The other approach to cropping regions is *implicitly* on featuremaps after feeding the full image to a CNN as done in Faster R-CNN [8], BagNet [47], and CNN pooling methods such as sum [46], BoW [50], VLAD [45, 51], Fisher vector [52]. In our paper we show that CNNs can encode the absolute position. This means that in contrast to explicitly cropping a region before the CNN, cropping a region after the CNN can include absolute position information, which impacts all implicit region cropping methods.

Robustness to image transformations. The semantic content of an image should be invariant to the accidental camera position. Robustness to such geometric transformation can be learned by adding them to the training set using data augmentation [53–57]. Instead of augmenting with random transformations there are geometric adverserial training methods [58–60] that intelligently add the most sensitive geometric transformations to the training data. Adding data to the training set by either data augmentation or adverserial training is a brute-force solution adding additional computation as the dataset grows.

Instead of adding transformed versions of the training data there are methods specifically designed to learn geometric transformations in an equivariant or invariant representation [61–63] where examples include rotation [64–68], scale [69–73] and other transformations [74–78]. Closely related is the observation that through subsequent pooling and subsampling in CNN layers translation equivariance is lost [79, 80]. In our paper, we also investigate the loss of translation equivariance, yet do not focus on pooling but instead show that convolutional layers can exploit image boundary effects to encode the absolute position which was also found independently by Islam *et al* [81].

Boundary effects. Boundary effects cause statistical biases in finitely sampled data [82, 83]. For image processing this is textbook material [37, 38], where boundary handling has applications in image restoration and deconvolutions [84–86]. Boundary handling in CNNs focuses on minimizing boundary effects by learning separate filters at the boundary [87], treating out of boundary pixels as missing values [88], circular convolutions for wrap-around input data such as 360° degree images [89] and minimizing distortions in 360° degree video [90]. We, instead, investigate how boundary effects can encode absolute spatial location.

Location information in CNNs. Several deep learning methods aim to exploit an absolute spatial location bias in the data [91, 92]. This bias stems from how humans take pictures where for example a sofa tends to be located on the bottom of the image while the sky tends to be at the top. Explicitly adding absolute spatial location information helps for patch matching [93, 94], generative modeling [95], semantic segmentation [92, 96], instance segmentation [97]. In this paper we do not add spatial location information. Instead, we do the opposite and show how to remove such absolute spatial location information from current CNNs.

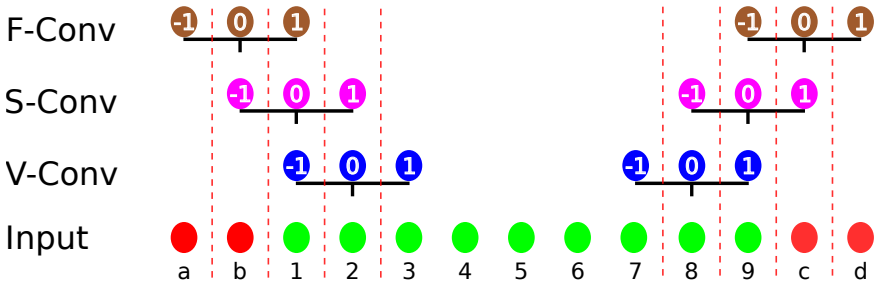


Figure 2.2: How convolution ignores positions close to the border. We show the first and the last position for three convolution types: Valid (V-Conv), Same (S-Conv) and Full (F-Conv) applied to an input with finite support (green) and border padding (red). Note that for V-conv, the blue filter at position 1 is never applied to the green input positions 1 and 2. For S-Conv, the pink filter position 1 is never applied to green input position 1. F-Conv has all filter values applied on the image.

Visual inductive priors for data efficiency. Adding visual inductive priors to deep learning increases data efficiency. Deep networks for image recognition benefit from a convolutional prior [98] and the architectural structure of a CNN with random weights already provides an inductive bias [99–101]. The seminial Scattering network [102] and its variants [103, 104] design a convolutional architecture to incorporate physical priors about image deformations. Other work shows that adding priors increases data efficiency by tying parameters [75], sharing rotation responses [67], and a prior scale-space filter basis [105]. In our paper we show that removing the ability of convolutional layers to exploit the absolute position improves translation equivariance and invariance which enforces the visual inductive prior of the convolution operator in deep learning.

2.3. HOW BOUNDARY EFFECTS ENCODE LOCATION

We explore common convolution types for boundary handling with their image padding variants and explore their equivariant and invariant properties. In Fig. 2.2 we illustrate the convolution types. For clarity of presentation we mostly focus on $d = 1$ dimensional convolutions in a single channel, although the analysis readily extends to the multi-dimensional multi-channel case. We use the term 'image' broadly and also includes feature maps.

Boundaries for convolution on finite samples. Let $\mathbf{x} \in \mathbb{R}^n$ be the 1-D single channel input image of size n and $\mathbf{f} \in \mathbb{R}^{2k+1}$ denote a 1-D single channel filter where for convenience we only consider odd sized filters of size $2k+1$. The output $y[t]$ for discrete convolution is

$$\mathbf{y}[t] = \sum_{j=-k}^{k} \mathbf{f}[j]\mathbf{x}[t-j]. \tag{2.1}$$

Images have finite support and require handling boundary cases, for example where $t - j < 0$ and $x[t-j]$ falls outside the defined image. Providing values outside the image boundary is commonly referred to as padding. We consider two cases. *Zero padding* assumes that all values outside of the images are zero. *Circular padding* wraps the image values on one side around to the other side to provide the missing values.

2.3.1. COMMON CONVOLUTIONS FOR BOUNDARY HANDLING

Valid convolution (V-Conv). V-Conv does not convolve across image boundaries. Thus, V-conv is a function $\mathbb{R}^n \rightarrow \mathbb{R}^{n-2k}$ where the output range of Eq. (2.1) is in the interval:

$$t \in [k+1, n-k]. \tag{2.2}$$

It only considers existing values and requires no padding. Note that the support of the output y has $2kd$ fewer elements than the input x, where d is the dimensionality of the image, *i.e*, the output image shrinks with k pixels at all boundaries.

Same convolution (S-Conv). S-Conv slides only the filter center on all existing image values. The output range of Eq. (2.1) is the same as the input domain; *i.e.* the interval:

$$t \in [1, n]. \tag{2.3}$$

The support of the output y is the same size as the support of the input x. Note that $2kd$ values fall outside the support of x, *i.e*, at each boundary there are k padding values required.

Full convolution (F-Conv). F-Conv applies each value in the filter on all values in the image. Thus, F-conv is a function $\mathbb{R}^n \rightarrow \mathbb{R}^{n+2k}$ where the output range of Eq. (2.1) is in the interval:

$$t \in [-k, n+k]. \tag{2.4}$$

The output support of y has $2kd$ more elements than the input x, *i.e*, the image grows with k elements at each boundary. Note that $4kd$ values fall outside of the support of the input x: At each boundary $2k$ padded values are required.

2.3.2. ARE ALL INPUT LOCATIONS EQUAL?

We investigate if convolution types are equally applied to all input position in an image. In Fig. 2.2 we illustrate the setting. To analyze if each location is equal, we modify Eq. (2.1) to count how often an absolute spatial position a in the input signal x is used in the convolution. The count $C(\cdot)$ sums over all input positions i where the convolution is applied,

$$C(a) = \sum_i \sum_{j=-k}^{k} [\![i = a - j]\!], \tag{2.5}$$

where $[\![\cdot]\!]$ are Iverson Brackets which evaluate to 1 if the expression in the brackets is true. Without boundary effects $C(a)$ always sums to $2k+1$ for each value of a.

When there are boundary effects, there will be differences. For V-Conv, the input locations i are determined by Eq. (2.2) and the equation becomes

$$C_V(a) = \sum_{i=k+1}^{n-k} \sum_{j=-k}^{k} [\![i = a - j]\!], \tag{2.6}$$

where i no longer sums over all values. Thus, for all locations in the input image the function $C_V(t)$ no longer sums to $2k+1$ as it does in Eq. (2.5), instead they sum to a

lower value. In fact, it reduces to

$$C_V(a) = \begin{cases} a & \text{if } a \in [1, 2k] \\ n - a + 1 & \text{if } a \in [n-2k, n] \\ 2k+1 & \text{Otherwise.} \end{cases} \tag{2.7}$$

This shows that for V-Conv there are absolute spatial locations where the full filter is not applied.

For S-Conv, where Eq. (2.3) defines the input, the count is

$$C_S(a) = \sum_{i=1}^{n} \sum_{j=-k}^{k} [\![i = a - j]\!], \tag{2.8}$$

where i sums over all values, and slides only the filter center over all locations. Thus, for S-Conv, when the locations are $a \leq k$ or $a \geq n-k$, the function $C_S(a)$ no longer sums to $2k+1$. This reduces to

$$C_S(a) = \begin{cases} a + k & \text{if } a \in [1, k] \\ n - a + (k+1) & \text{if } a \in [n-k, n] \\ 2k+1 & \text{Otherwise.} \end{cases} \tag{2.9}$$

This means that also for S-Conv there are absolute spatial locations where the full filter is not applied.

S-Conv with circular padding 'wraps around' the image and uses the values on one side of the image to pad the border on the other side. Thus, while for S-Conv, Eq. (2.9) holds for the absolute position i, it is by using circular padding that the *value* $x[i]$ at position i is exactly wrapped around to the positions where the filter values were not applied. Hence, circular padding equalizes all responses, albeit at the other side of the image. Zero padding, in contrast, will have absolute spatial locations where filter values are never applied.

For F-Conv, in Eq. (2.4), the counting equation becomes

$$C_F(a) = \sum_{i=-k}^{n+k} \sum_{j=-k}^{k} [\![i = a - j]\!]. \tag{2.10}$$

F-Conv sums the filter indices over all indices in the image and thus, as in Eq. (2.5), all locations i sum to $2k+1$ and thus no locations are left out.

We conclude that V-Conv is the most sensitive to exploitation of the absolute spatial location. S-Conv with zero padding is also sensitive to location exploitation. S-Conv with circular padding is not sensitive, yet involves wrapping values around to the other side, which may introduce semantic artifacts. F-Conv is not sensitive to location information. In Fig. 2.3 we give an example of all convolution types and how they can learn absolute spatial position.

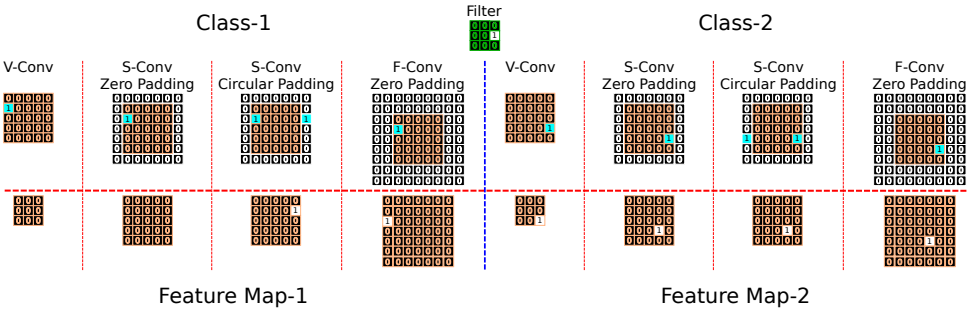


Figure 2.3: A 2D Example where a pixel on the top left input (Class-1) and the same pixel on the bottom-right input (Class-2) can be classified using convolution. Comparing the output of 4 convolution types shows that V-Conv and S-Conv for Class-1 can no longer detect the pixel, while Class-2 still has the pixel. S-Conv with circular padding and F-Conv always retain the pixel value.

2.4. EXPERIMENTS

Implementation details for Full Convolution. For standard CNNs implementing F-Conv is trivially achieved by simply changing the padding size. For networks with residual connections, we add additional zero padding to the residual output to match the spatial size of the feature map.

2.4.1. EXP 1: HOW FAR FROM THE IMAGE BOUNDARY CAN ABSOLUTE LOCATION BE EXPLOITED?

CNNs can encode absolute position by exploiting boundary effects. In this experiment we investigate how far from the boundary these effects can occur. Can absolute position be encoded only close to the boundary or also far away from the boundary? To answer this question we revisit the location classification setting in Fig. 2.1 while adding an increasingly large border all around the image until location can no longer be classified. In Fig. 2.4 we show the setting.

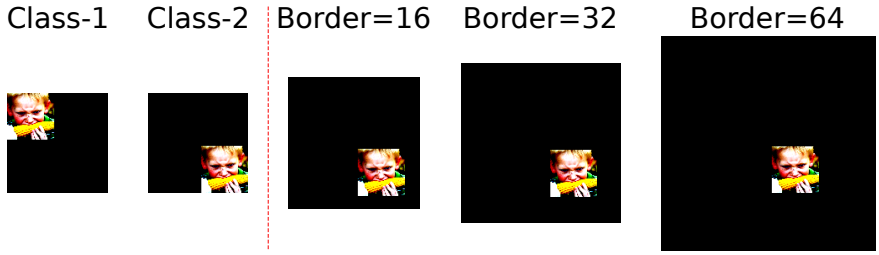


Figure 2.4: **Exp 1:** Example images. Evaluating how far from the image boundary absolute location can be exploited. The task is to classify the location of a 56x56 resized Imagenet image placed in the top-left (class-1) and bottom-right (class-2), see also Fig. 2.1. We add a border on all 4 sides of the image, where we increase the border size until location can no longer be classified.

We randomly pick 3,000 samples from ImageNet validation set, resize them to 56x56 and distribute them equally in a train/val/test set. For each of the 3k samples we create

two new images (so, 2,000 images in each of the 3 train/val/test sets) by taking a black 112x112 image and placing the resized ImageNet sample in the top-left corner (class-1) and in the bottom-right corner (class-2), see Fig. 2.1. To evaluate the distance from the boundary we create 7 versions by adding a black border of size $\in \{0, 16, 32, 64, 128, 256, 512\}$ on all 4 sides of the 112x112 image, see Fig. 2.4 for examples.

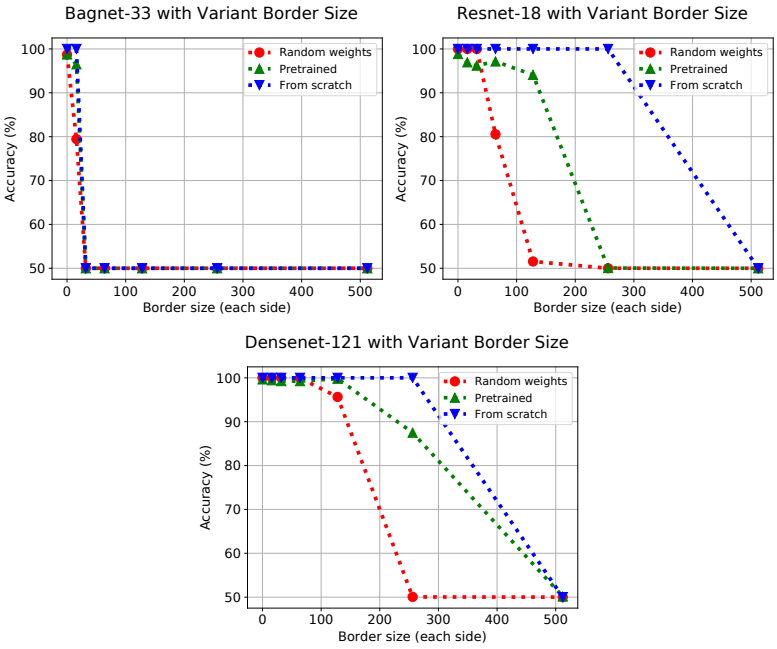


Figure 2.5: **Exp 1:** Evaluating a BagNet-33 [47] (left), a ResNet-18 [3] (middle) and a DenseNet-121 [33] (right) on how far from the boundary absolute location can be exploited, see Fig. 2.4. The x-axis is the border size added to all 4 sides of the image and the y-axis is accuracy. All models can classify absolute position. The small RF of the BagNet allows for classification close to the border. The ResNet-18 and DenseNet-121 have larger RFs and can classify location far from the boundary. Random convolutional weights stay relatively close to the boundary while training on ImageNet learns filters that can go further. Training from scratch does best. Note that the most distant location from an image boundary for a kxk image is a border size of $k/2$, *i.e*, a border size of 128 corresponds to a 256x256 image.

We evaluate three networks with varying receptive field size. BagNet-33 [47] is a ResNet variant where the receptive field is constrained to be 33x33 pixels. ResNet-18 [3] is a medium sized network, while a DenseNet-121 [33] is slightly larger. We evaluate three settings: (i) trained completely from scratch to see how well it can do; (ii) randomly initialized with frozen convolution weights to evaluate the architectural bias for location classification; (iii) ImageNet pre-trained with frozen convolution weights to evaluate the location classification capacity of a converged realistic model used in a typical image classification setting.

Results in Fig. 2.5 show that all settings for BagNet, ResNet and DenseNet can classify absolute position. Random weights can do it for locations relatively close to the boundary. Surprisingly, the pretrained models have learned filters on ImageNet that can

classify position further away from the boundary as compared to random initialization. The models trained from scratch can classify absolute position the furthest away from the boundary. The BagNet fails for locations far from the boundary. Yet, the medium-sized ResNet-18 can still classify locations of 128 pixels away from the boundary, which fully captures ImageNet as for 224x224 images the most distant pixel is only 112 pixels from a boundary. We conclude that absolute location can even be exploited far from the boundary.

2.4.2. EXP 2: BORDER HANDLING VARIANTS

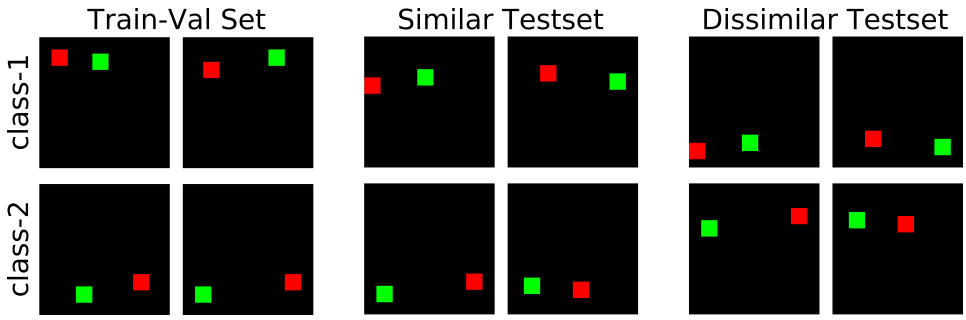


Figure 2.6: **Exp 2:** Example images of the Red-Green two class classification dataset for evaluating exploitation of absolute position. The upper row of images is class 1: Red-to-the-left-of-Green. The lower row of images is class 2: Green-to-the-left-of-Red. The Similar Testset is matching the Train-Val set in absolute location: Class 1 at the top and class 2 at the bottom. The Dissimilar testset is an exact copy of the Similar testset where absolute location is swapped between classes: Class 1 at the bottom, Class 2 at the top. If absolute location plays no role then classification on the Similar Testset would perform equal to the Dissimilar Testset.

Border handling is the key to absolute location coding. Here we evaluate the effect of various border handling variants on absolute location exploitation. To do so, we create an image classification task unrelated to the absolute position and introduce a location bias which should have no effect on translation invariant architectures.

We construct the Red-Green data set for binary image classification of the relative order of colored blocks of 4x4 on a black 32x32 image. Class 1 has Red to the left of Green; class 2 has Green to the left of Red, see Fig. 2.6. The classification task is unrelated to the absolute position. We introduce a vertical absolute position bias by placing class 1 on the top of the image (8 pixels from the top, on average), and class 2 on the bottom (8 pixels from the bottom, on average). We then construct two test sets, one with similar absolute location bias, and a dissimilar test set where the location bias switched: class 1 at the bottom and class 2 on top, see Fig. 2.6.

The train set has 2,000 images, the validation and test sets each have 1,000 images. Experiments are repeated 10 times with different initialization of the networks. A 4-layer fully convolutional deep network is used for evaluation. The first two layers have 32 filters and last two layers 64 filter followed by global max pooling. Sub-sampling for layers 2, 3, 4 uses stride 2 convolution.

We evaluate the border handling of Section 2.3. *V-Conv* uses only existing image values and no padding. For *S-Conv* we evaluate zero and circular padding. *F-Conv* has zero

Type	Pad	Similar test	Dissimilar test
V-Conv	-	**100.0 ± 0.0**	0.2 ± 0.1
S-Conv	Zero	99.8 ± 0.1	8.4 ± 0.7
S-Conv	Circ	73.7 ± 1.0	73.7 ± 1.0
F-Conv	Zero	89.7 ± 0.5	**89.7 ± 0.5**

Table 2.1: **Exp 2:** Accuracy on the Red-Green dataset shown in Fig. 2.6. Type is the convolution type, pad is how padding is done. Results are given on the Similar test set with matching absolute positions and the Dissimilar test set with an absolute position mismatch. Stddevs are computed by 10 repeats. *Valid* and *same*-zero exploit location and do poorly on the Dissimilar test set. *Same*-circ is translation invariant yet invents disturbing new content. *Full*-zero is translation invariant, doing well on both test sets.

padding. Results are in Table 2.1. *V-Conv* and *S-Conv*-zero have the best accuracy on the Similar test set, yet they exploit the absolute location bias and perform poorly on the Dissimilar test set, where *V-Conv* relies exclusively on location and confuses the classes completely. *S-Conv*-circ and *F-Conv* perform identical on the Similar and Dissimilar test sets; they are translation invariant and thus cannot exploit the absolute location bias. *F-Conv* does better than *S-Conv*-circ because circular padding introduces new content. *F-Conv* does best on both test sets as it is translation invariant and does not introduce semantic artifacts.

2.4.3. EXP 3: SENSITIVITY TO IMAGE SHIFTS

Does removing absolute location as a feature lead to robustness to location shifts? We investigate the effect of image shifts at test time on CNN output for various architectures on a subset of ImageNet. We train four different architectures from scratch with S-Conv and F-Conv: Resnet 18, 34, 50 and 101. To speed up training from scratch, we use 20% of the full ImageNet and take the 200 classes from [106] which is still large but 5x faster to train. To evaluate image shifts we follow the setting of BlurPool [80], which investigates the effect of pooling on CNN translation equivariance. As BlurPool improves equivariance, we also evaluate the effect of BlurPool Tri-3 [80].

Diagonal Shift. We train the network with the usual central crop. Each testing image is diagonally shifted starting from the top-left corner towards the bottom-right corner. We shift 64 times 1 pixel diagonally. Accuracy is evaluated for each pixel shift and averaged over the full test set.

Consistency. We measure how often the classification output of a model is the same for a pair of randomly chosen diagonal shifts between 1 and 64 pixels [80]. We evaluate each test image 5 times and average the results.

Results are given in Table 2.2. For each architecture, using F-Conv improves both the classification performance and the consistency of all the models. The highest classification accuracy gain between S-Conv and F-Conv is 3.6% and the best consistency gain is 2.49% with Resnet-34. BlurPool makes S-Convs more robust to diagonal shifts and increase consistency. When F-Conv and BlurPool are combined, the accuracy on diagonal shifting and consistency are improved further. Resnet-34 (F+BlurPool) obtains more 4.85% of accuracy and 3.91% of consistency compared to the S-Conv baseline. If we compare each Resnet architecture, the deepest model of the experiment, Resnet-101,

Diagonal Shift	S-Conv	F-Conv	S+BlurPool	F+BlurPool
RN18	79.43	82.74	81.96	**83.95**
RN34	82.06	85.66	83.73	**86.91**
RN50	86.36	87.92	87.50	**88.93**
RN101	86.95	87.78	88.22	**88.73**
Consistency	S-Conv	F-Conv	S+BlurPool	F+BlurPool
RN18	86.43	88.38	88.32	**90.03**
RN34	87.62	90.12	89.21	**91.53**
RN50	90.21	91.36	91.68	**92.75**
RN101	90.76	91.71	92.36	**92.86**

Table 2.2: **Exp 3:** Diagonal shift and consistency result for different Resnet architectures. S+BlurPool represents S-Convs with BlurPool Tri-3. Similarly, F+BlurPool corresponds the combination of F-Conv and BlurPool. In the most cases, F-Conv outperforms S-Conv and S+BlurPool (except for Resnet-101) in terms of diagonal shifting accuracy on testing set. Similar trend can be seen for consistency experiment, yet for Resnet-50 and Resnet-101, S+BlurPool has more consistent outputs. F+BlurPool achieves the highest score for both cases with all the architectures.

improves the least, both for classification and consistency. Resnet-101 has more filters and parameters and it can learn many more varied filters than other models. By this, it can capture many variants of location of objects and thus the gap between methods for Resnet-101 are smaller.

2.4.4. EXP 4: DATA EFFICIENCY

Does improving equivariance and invariance for the inductive convolutional prior lead to benefits for smaller data sets? We evaluate S-Conv and F-Conv with the same random initialization seed for two different settings: Image classification and image patch matching.

Image classification. We evaluate ResNet-50 classification accuracy for various training set sizes of the 1,000 classes in ImageNet. We vary the training set size as 50, 100, 250, 500, and all images per class.

Patch matching. We use HardNet [107] and use FPR (false positive rate) at 0.95 true positive recall as an evaluation metric (lower is better). We evaluate on 3 common patch matching datasets (Liberty, Notre Dame and Yosemite) from Brown dataset [108] where the model is trained on one set and tested on the other two sets. Hardnet uses triplets loss and we vary the training set size as 50k, 100k, 250k, 500k triplet patches. Each test set has 100k triplet patches.

Results are given in Fig. 2.7. For both image classification as for patch matching S-Conv and F-Conv perform similar for a large amount of training data. Yet, when reducing the number of training samples there is a clear improvement for F-Conv. For ImageNet with only 50 samples per class S-Conv scores 26.4% and F-Conv scores 31.1%, which is a relative improvement of 17.8%. For patch matching, S-Conv scores 0.145 and F-Conv 0.083 which is a relative improvement of 75%. Clearly, removing absolute location

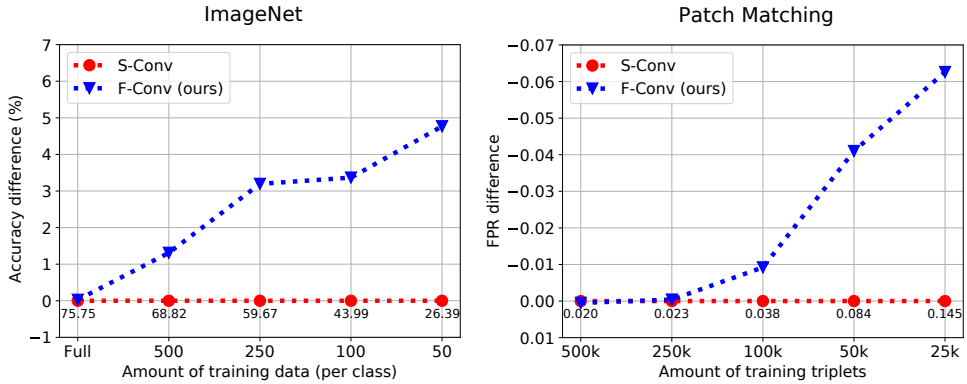


Figure 2.7: **Exp 4:** Data efficiency experiments. We reduce the amount of training data per class for the 1,000 classes of Imagenet for image classification and full Liberty, Notre Dame and Yosemite for patch matching. F-Conv outperforms S-Conv in both modality with smaller data size. (left) The Imagenet plot demonstrates the obtained accuracy difference when the number of data samples per class. The difference between F-Conv and S-Conv increases when the sample size decreases. (right) Correspondingly, F-Conv results in a performance increase for patch matching.

	UCF101		HMDB51	
	Baseline (S-Conv)	Ours (F-Conv)	Baseline (S-Conv)	Ours (F-Conv)
RN-18	38.6	40.6*	16.1	19.3
RN-34	37.0	46.9	15.2	18.3
RN-50	36.2	44.1	14.3	19.0

Table 2.3: **Exp 5:** Action recognition with 3D Resnet-18, 34 and 50 by using S-Conv and F-Conv methods. F-Conv outperforms S-Conv on UCF101 and HMDB51 datasets. S-Conv obtains its best result with the most shallow network, Resnet-18, however F-Conv still improves the results even the model becomes bigger. (*) Enabling Fconv also in the temporal dimension improves the performance by 1.6%.

improves data efficiency.

2.4.5. EXP 5: SMALL DATASETS

Here we evaluate if the improved data efficiency generalize to two small datasets for action recognition. We select small sized data sets where training from scratch gives significantly worse results due to overfitting and the common practice is pre-training on a huge third party dataset. We compare the standard S-Conv with the proposed F-Conv where both methods are trained from scratch.

Action Recognition. We evaluate on two datasets: UCF101 [109] with 13k video clips from 101 action classes and HMDB51 [110] with 51 action classes and around 7k annotated video clips. We evaluate three 3D Resnet architectures [16], Resnet-18, 34 and 50.

We show results in Table 2.3. F-Conv models outperform the S-Conv models. Interestingly, in UCF101 experiment, the baseline performance decreased by 2.4% from Resnet-18 to Resnet-50; however, F-Convs still continue to improve the performance

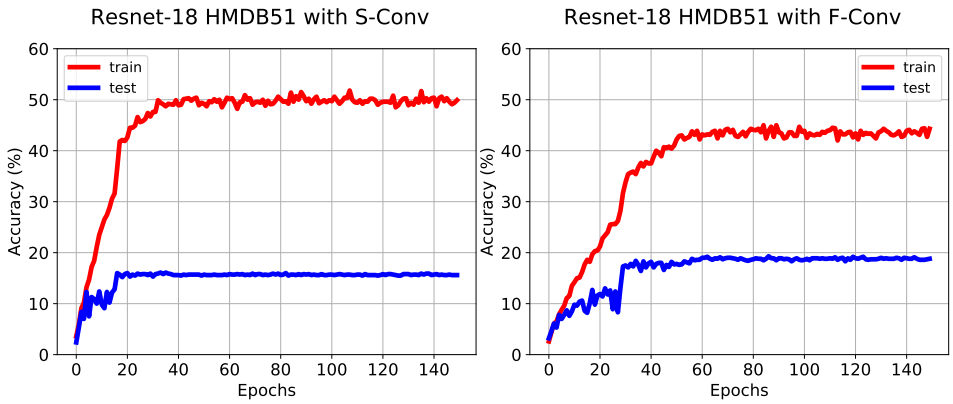


Figure 2.8: **Exp 5:** Training curves for 3D Resnet-18 S-Conv (left) and F-Conv (right) with HMDB51 dataset. Because the dataset is small, both models overfit. F-Conv achieves relatively 38.8% less overfitting than S-Conv.

by 3.6% for same architectures. Besides, enabling Fconv in the temporal dimension increases the performance of Resnet-18 by 1.6%. According to Kensho et al [16] a 3D Resnet-18 overfits with UCF101 and HMDB51 which we confirm, yet F-Conv we overfit less than S-Conv. In Fig. 2.8, the difference between train and test of a 3D Resnet18 with S-Conv is 35.69%, however F-Conv has 25.7% overfitting. Similarly, S-Conv is relatively 41% more overfitted than F-Conv in Fig. 2.9. Consequently, both methods overfit due to the number of parameter and the lack of data.

2.5. LIMITATIONS AND CONCLUSION

One limitation of our method is the extra computation required for padding. There is no extra cost of using circular padding instead of zero padding. For using F-Conv instead of S-Conv, the costs are similar to using S-Conv instead of V-Conv, and we found a Resnet-50 with F-Conv 15% slower to train on Imagenet.

Note that if absolute spatial location is truly discriminative between classes, it *should* be exploited [111], and not removed. For many internet images with a human photographer, there will be a location bias as humans tend to take pictures with the subject in the center, sofas on the bottom, and the sky up. The difficulty lies in having deep networks not exploit spurious location correlations due to lack of data. Addressing lack of data samples by sharing parameters over locations through added convolutions in deep networks is a wonderfully regularizer and we believe that convolutional layers should truly be translation equivariant.

To conclude, we show that in contrary to popular belief, convolutional layers can encode the absolute spatial location in an image. With the strong presence of the convolution operator in deep learning this insight is relevant to a broad audience. We analyzed how boundary effects allow for ignoring certain parts of the image. We evaluated existing networks and demonstrated that their large receptive field makes absolute spatial location coding available all over the image. We demonstrate that removing spatial

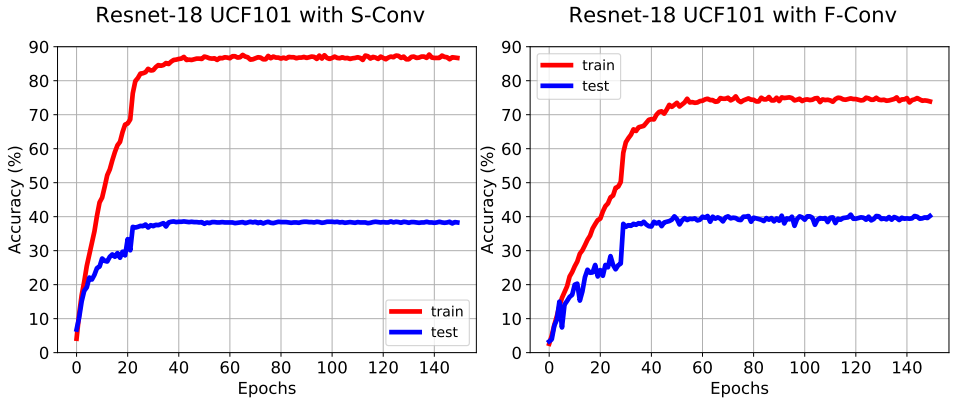


Figure 2.9: **Exp 5:** Training curves for 3D Resnet-18 S-Conv (left) and F-Conv (right) with UCF101 dataset. Both models overfit, but S-Conv has higher difference between training and testing results (49.1%). F-Conv has 34.8% of gap and thus overfits less.

location as a feature increases the stability to image shifts and improves the visual inductive prior of the convolution operator which leads to increased accuracy in the low-data regime and small datasets which we demonstrate for ImageNet image classification, image patch matching, and two video classification data sets.

REFERENCES

[1] O. S. Kayhan and J. C. v. Gemert, *On translation invariance in cnns: Convolutional layers can exploit absolute spatial location,* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks,* in *Advances in neural information processing systems* (2012) pp. 1097–1105.

[3] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016) pp. 770–778.

[4] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition,* arXiv preprint arXiv:1409.1556 (2014).

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going deeper with convolutions,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015) pp. 1–9.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, *Ssd: Single shot multibox detector,* in *European conference on computer vision* (Springer, 2016) pp. 21–37.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016) pp. 779–788.

[8] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks,* in *Advances in neural information processing systems* (2015) pp. 91–99.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, *Mask r-cnn,* in *Proceedings of the IEEE international conference on computer vision* (2017) pp. 2961–2969.

[10] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, *Panoptic segmentation,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019) pp. 9404–9413.

[11] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation,* in *International Conference on Medical image computing and computer-assisted intervention* (Springer, 2015) pp. 234–241.

[12] J. L. Long, N. Zhang, and T. Darrell, *Do convnets learn correspondence?* in *Advances in Neural Information Processing Systems* (2014) pp. 1601–1609.

[13] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, *Matchnet: Unifying feature and metric learning for patch-based matching,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015) pp. 3279–3286.

[14] S. Zagoruyko and N. Komodakis, *Learning to compare image patches via convolutional neural networks,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015) pp. 4353–4361.

[15] J. Carreira and A. Zisserman, *Quo vadis, action recognition? a new model and the kinetics dataset,* in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017) pp. 6299–6308.

[16] K. Hara, H. Kataoka, and Y. Satoh, *Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?* in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2018) pp. 6546–6555.

[17] K. Simonyan and A. Zisserman, *Two-stream convolutional networks for action recognition in videos,* in *Advances in neural information processing systems* (2014) pp. 568–576.

[18] L. A. Gatys, A. S. Ecker, and M. Bethge, *Image style transfer using convolutional neural networks,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016) pp. 2414–2423.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets,* in *Advances in neural information processing systems* (2014) pp. 2672–2680.

[20] D. P. Kingma and P. Dhariwal, *Glow: Generative flow with invertible 1x1 convolutions,* in *Advances in Neural Information Processing Systems* (2018) pp. 10215–10224.

[21] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, *Convolutional neural networks for speech recognition,* IEEE/ACM Transactions on audio, speech, and language processing **22**, 1533 (2014).

[22] Y. LeCun, Y. Bengio, *et al., Convolutional networks for images, speech, and time series,* The handbook of brain theory and neural networks **3361**, 1995 (1995).

[23] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, *Wavenet: A generative model for raw audio,* arXiv preprint arXiv:1609.03499 (2016).

[24] K. Choi, G. Fazekas, M. Sandler, and K. Cho, *Convolutional recurrent neural networks for music classification,* in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2017) pp. 2392–2396.

[25] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, *et al., Cnn architectures for large-scale audio classification,* in *2017 ieee international conference on acoustics, speech and signal processing (icassp)* (IEEE, 2017) pp. 131–135.

[26] J. Salamon and J. P. Bello, *Deep convolutional neural networks and data augmentation for environmental sound classification,* IEEE Signal Processing Letters **24**, 279 (2017).

[27] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, *On the properties of neural machine translation: Encoder–decoder approaches,* Syntax, Semantics and Structure in Statistical Translation , 103 (2014).

[28] C. Dos Santos and M. Gatti, *Deep convolutional neural networks for sentiment analysis of short texts,* in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (2014) pp. 69–78.

[29] S. Lai, L. Xu, K. Liu, and J. Zhao, *Recurrent convolutional neural networks for text classification,* in *Twenty-ninth AAAI conference on artificial intelligence* (2015).

[30] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, *Spectral networks and locally connected networks on graphs,* in *International Conference on Learning Representations (ICLR2014), CBLS, April 2014* (2014) pp. http–openreview.

[31] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, *Convolutional networks on graphs for learning molecular fingerprints,* in *Advances in neural information processing systems* (2015) pp. 2224–2232.

[32] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, *Modeling relational data with graph convolutional networks,* in *European Semantic Web Conference* (Springer, 2018) pp. 593–607.

[33] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, *Densely connected convolutional networks,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017) pp. 4700–4708.

[34] M. Lin, Q. Chen, and S. Yan, *Network in network,* in *ICLR* (2013).

[35] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, *Striving for simplicity: The all convolutional net,* in *ICLR (workshop track)* (2015).

[36] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, *Aggregated residual transformations for deep neural networks,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017) pp. 1492–1500.

[37] B. Jahne, *Digital image processing*, Vol. 4 (Springer Berlin, 2005).

[38] G. Strang and T. Nguyen, *Wavelets and filter banks* (SIAM, 1996).

[39] A. Araujo, W. Norris, and J. Sim, *Computing receptive fields of convolutional neural networks,* Distill (2019), 10.23915/distill.00021, https://distill.pub/2019/computing-receptive-fields.

[40] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, *Gradient-based learning applied to document recognition,* Proceedings of the IEEE **86**, 2278 (1998).

[41] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014) pp. 580–587.

[42] L. Hou, D. Samaras, T. M. Kurc, Y. Gao, J. E. Davis, and J. H. Saltz, *Patch-based convolutional neural network for whole slide tissue image classification,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 2424–2433.

[43] A. Sharma, X. Liu, X. Yang, and D. Shi, *A patch-based convolutional neural network for remote sensing image classification,* Neural Networks **95**, 19 (2017).

[44] J. Zbontar, Y. LeCun, *et al.*, *Stereo matching by training a convolutional neural network to compare image patches.* Journal of Machine Learning Research **17**, 2 (2016).

[45] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, *Netvlad: Cnn architecture for weakly supervised place recognition,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016) pp. 5297–5307.

[46] A. Babenko and V. Lempitsky, *Aggregating local deep features for image retrieval,* in *Proceedings of the IEEE international conference on computer vision* (2015) pp. 1269–1277.

[47] W. Brendel and M. Bethge, *Approximating cnns with bag-of-local-features models works surprisingly well on imagenet,* in *ICLR* (2019).

[48] A. Richard and J. Gall, *A bag-of-words equivalent recurrent neural network for action recognition,* Computer Vision and Image Understanding **156**, 79 (2017).

[49] T. Shen, Y. Huang, and Z. Tong, *Facebagnet: Bag-of-local-features model for multi-modal face anti-spoofing,* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2019).

[50] N. Passalis and A. Tefas, *Learning bag-of-features pooling for deep convolutional neural networks,* in *Proceedings of the IEEE international conference on computer vision* (2017) pp. 5755–5763.

[51] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, *Multi-scale orderless pooling of deep convolutional activation features,* in *European conference on computer vision* (Springer, 2014) pp. 392–407.

[52] M. Cimpoi, S. Maji, I. Kokkinos, and A. Vedaldi, *Deep filter banks for texture recognition, description, and segmentation,* International Journal of Computer Vision **118**, 65 (2016).

[53] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, *Autoaugment: Learning augmentation strategies from data,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019) pp. 113–123.

[54] A. Fawzi, H. Samulowitz, D. Turaga, and P. Frossard, *Adaptive data augmentation for image classification,* in *2016 IEEE International Conference on Image Processing (ICIP)* (Ieee, 2016) pp. 3688–3692.

[55] A. Hernández-García and P. König, *Data augmentation instead of explicit regularization,* arXiv preprint arXiv:1806.03852 (2018).

[56] D. Ho, E. Liang, X. Chen, I. Stoica, and P. Abbeel, *Population based augmentation: Efficient learning of augmentation policy schedules,* in *International Conference on Machine Learning* (2019) pp. 2731–2741.

[57] E. Kauderer-Abrams, *Quantifying translation-invariance in convolutional neural networks,* arXiv preprint arXiv:1801.01450 (2017).

[58] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, *Exploring the landscape of spatial robustness,* in *International Conference on Machine Learning* (2019) pp. 1802–1811.

[59] A. Fawzi and P. Frossard, *Manitest: Are classifiers really invariant?* in *British Machine Vision Conference (BMVC)*, CONF (2015).

[60] C. Kanbak, S.-M. Moosavi-Dezfooli, and P. Frossard, *Geometric robustness of deep networks: Analysis and improvement,* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).

[61] A. Bietti and J. Mairal, *Group invariance, stability to deformations, and complexity of deep convolutional representations,* The Journal of Machine Learning Research **20**, 876 (2019).

[62] R. Kondor and S. Trivedi, *On the generalization of equivariance and convolution in neural networks to the action of compact groups,* in *International Conference on Machine Learning* (2018) pp. 2752–2760.

[63] K. Lenc and A. Vedaldi, *Understanding image representations by measuring their equivariance and equivalence,* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

[64] S. Dieleman, J. De Fauw, and K. Kavukcuoglu, *Exploiting cyclic symmetry in convolutional neural networks,* in *ICML* (JMLR. org, 2016) pp. 1889–1898.

[65] D. Marcos, M. Volpi, N. Komodakis, and D. Tuia, *Rotation equivariant vector field networks,* in *ICCV* (2017) pp. 5048–5057.

[66] M. Weiler, F. A. Hamprecht, and M. Storath, *Learning steerable filters for rotation equivariant cnns,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018) pp. 849–858.

[67] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, *Harmonic networks: Deep translation and rotation equivariance,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017) pp. 5028–5037.

[68] Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao, *Oriented response networks,* in *CVPR* (2017) pp. 519–528.

[69] D. Marcos, B. Kellenberger, S. Lobry, and D. Tuia, *Scale equivariance in cnns with vector fields,* arXiv preprint arXiv:1807.11783 (2018).

[70] I. Sosnovik, M. Szmaja, and A. Smeulders, *Scale-equivariant steerable networks,* arXiv preprint arXiv:1910.11093 (2019).

[71] Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen, *Self-supervised scale equivariant network for weakly supervised semantic segmentation,* arXiv preprint arXiv:1909.03714 (2019).

[72] D. E. Worrall and M. Welling, *Deep scale-spaces: Equivariance over scale,* arXiv preprint arXiv:1905.11697 (2019).

[73] Y. Xu, T. Xiao, J. Zhang, K. Yang, and Z. Zhang, *Scale-invariant convolutional neural networks,* arXiv preprint arXiv:1411.6369 (2014).

[74] T. Cohen and M. Welling, *Group equivariant convolutional networks,* in *International conference on machine learning* (2016) pp. 2990–2999.

[75] R. Gens and P. M. Domingos, *Deep symmetry networks,* in *Advances in neural information processing systems* (2014) pp. 2537–2545.

[76] J. F. Henriques and A. Vedaldi, *Warped convolutions: Efficient invariance to spatial transformations,* in *ICML* (JMLR. org, 2017) pp. 1461–1469.

[77] D. Laptev, N. Savinov, J. M. Buhmann, and M. Pollefeys, *Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks,* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016) pp. 289–297.

[78] K. Sohn and H. Lee, *Learning invariant representations with local transformations,* in *Proceedings of the 29th International Coference on International Conference on Machine Learning* (Omnipress, 2012) pp. 1339–1346.

[79] A. Azulay and Y. Weiss, *Why do deep convolutional networks generalize so poorly to small image transformations?* arXiv preprint arXiv:1805.12177 (2018).

[80] R. Zhang, *Making convolutional networks shift-invariant again,* in *International Conference on Machine Learning* (2019) pp. 7324–7334.

[81] M. A. Islam, S. Jia, and N. D. Bruce, *How much position information do convolutional neural networks encode?* in *ICLR* (2019).

[82] D. A. Griffith, *The boundary value problem in spatial statistical analysis,* Journal of regional science **23**, 377 (1983).

[83] D. A. Griffith and C. G. Amrhein, *An evaluation of correction techniques for boundary effects in spatial statistical analysis: traditional methods,* Geographical Analysis **15**, 352 (1983).

[84] F. Aghdasi and R. K. Ward, *Reduction of boundary artifacts in image restoration,* IEEE Transactions on Image Processing **5**, 611 (1996).

[85] R. Liu and J. Jia, *Reducing boundary artifacts in image deconvolution,* in *2008 15th IEEE International Conference on Image Processing* (IEEE, 2008) pp. 505–508.

[86] S. J. Reeves, *Fast image restoration without boundary artifacts,* IEEE Transactions on image processing **14**, 1448 (2005).

[87] C. Innamorati, T. Ritschel, T. Weyrich, and N. J. Mitra, *Learning on the edge: Investigating boundary filters in cnns,* International Journal of Computer Vision , 1.

[88] G. Liu, K. J. Shih, T.-C. Wang, F. A. Reda, K. Sapra, Z. Yu, A. Tao, and B. Catanzaro, *Partial convolution based padding,* arXiv preprint arXiv:1811.11718 (2018).

[89] S. Schubert, P. Neubert, J. Pöschmann, and P. Pretzel, *Circular convolutional neural networks for panoramic images and laser data,* in *2019 IEEE Intelligent Vehicles Symposium (IV)* (IEEE, 2019) pp. 653–660.

[90] H.-T. Cheng, C.-H. Chao, J.-D. Dong, H.-K. Wen, T.-L. Liu, and M. Sun, *Cube padding for weakly-supervised saliency prediction in 360 videos,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018) pp. 1420–1429.

[91] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski, *An intriguing failing of convolutional neural networks and the coordconv solution,* in *Advances in Neural Information Processing Systems* (2018) pp. 9605–9616.

[92] Z. Wang and O. Veksler, *Location augmentation for cnn,* arXiv preprint arXiv:1807.07044 (2018).

[93] Z. Luo, T. Shen, L. Zhou, S. Zhu, R. Zhang, Y. Yao, T. Fang, and L. Quan, *Geodesc: Learning local descriptors by integrating geometry constraints,* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018).

[94] A. Mukundan, G. Tolias, and O. Chum, *Explicit spatial encoding for deep local descriptors,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019) pp. 9394–9403.

[95] N. Watters, L. Matthey, C. P. Burgess, and A. Lerchner, *Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes,* in *ICLR workshop* (2019).

[96] X. He, Z. Mo, Q. Chen, A. Cheng, P. Wang, and J. Cheng, *Location-aware upsampling for semantic segmentation,* (2019), arXiv:1911.05250 [cs.CV] .

[97] D. Neven, B. D. Brabandere, M. Proesmans, and L. V. Gool, *Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019) pp. 8837–8845.

[98] G. Urban, K. J. Geras, S. E. Kahou, O. Aslan, S. Wang, A. Mohamed, M. Philipose, M. Richardson, and R. Caruana, *Do deep convolutional nets really need to be deep and convolutional?* in *ICLR* (2016).

[99] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, *What is the best multi-stage architecture for object recognition?* in *2009 IEEE 12th international conference on computer vision* (IEEE, 2009) pp. 2146–2153.

[100] A. M. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Y. Ng, *On random weights and unsupervised feature learning.* in *ICML*, Vol. 2 (2011) p. 6.

[101] D. Ulyanov, A. Vedaldi, and V. Lempitsky, *Deep image prior,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018) pp. 9446–9454.

[102] J. Bruna and S. Mallat, *Invariant scattering convolution networks,* IEEE transactions on pattern analysis and machine intelligence **35**, 1872 (2013).

[103] E. Oyallon and S. Mallat, *Deep roto-translation scattering for object classification,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015) pp. 2865–2873.

[104] E. Oyallon, S. Zagoruyko, G. Huang, N. Komodakis, S. Lacoste-Julien, M. B. Blaschko, and E. Belilovsky, *Scattering networks for hybrid representation learning,* IEEE transactions on pattern analysis and machine intelligence (2018).

[105] J.-H. Jacobsen, J. van Gemert, Z. Lou, and A. W. Smeulders, *Structured receptive fields in cnns,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 2610–2619.

[106] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, *Natural adversarial examples,* arXiv preprint arXiv:1907.07174 (2019).

[107] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, *Working hard to know your neighbor's margins: Local descriptor learning loss,* in *Advances in Neural Information Processing Systems* (2017) pp. 4826–4837.

[108] M. Brown and D. G. Lowe, *Automatic panoramic image stitching using invariant features,* International Journal of Computer Vision **74**, 59 (2007).

[109] K. Soomro, A. R. Zamir, and M. Shah, *UCF101: A dataset of 101 human actions classes from videos in the wild,* CoRR **abs/1212.0402** (2012), arXiv:1212.0402 .

[110] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, *Hmdb: a large video database for human motion recognition,* in *2011 International Conference on Computer Vision* (IEEE, 2011) pp. 2556–2563.

[111] J. C. Van Gemert, *Exploiting photographic style for category-level image classification by generalizing the spatial pyramid,* in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval* (2011) pp. 1–8.

3

EVALUATING CONTEXT FOR DEEP OBJECT DETECTORS

There is no learning without remembering.

Socrates

This chapter has been published as:
O. S. Kayhan and J. C. van Gemert, *Evaluating the context for deep object detectors*, arXiv e-prints (2021).

ABSTRACT

Which object detector is suitable for your context sensitive task? Deep object detectors exploit scene context for recognition differently. In this paper, we group object detectors into 3 categories in terms of context use: no context by cropping the input (RCNN), partial context by cropping the featuremap (two-stage methods) and full context without any cropping (single-stage methods). We systematically evaluate the effect of context for each deep detector category. We create a fully controlled dataset for varying context and investigate the context for deep detectors. We also evaluate gradually removing the background context and the foreground object on MS COCO. We demonstrate that single-stage and two-stage object detectors can and will use the context by virtue of their large receptive field. Thus, choosing the best object detector may depend on the application context. The code[1] and dataset will be available.

[1] https://github.com/oskyhn/Detectors-Context

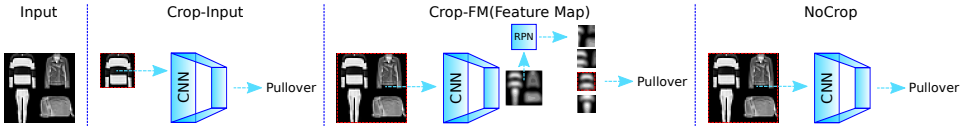


Figure 3.1: ROI handling in deep object detectors. Crop-Input (no context) crops the ROI from the input image before the CNN. Crop-FM (partial context) inputs the full image and crops the ROIs from the featuremap. NoCrop (full context) has a single-stage and uses the full image as input to the CNN.

3.1. INTRODUCTION

Objects are rarely photographed alone. An image may contain several other objects or varying scene background. This background context may correlate with the object and thus possibly exploited by a learned object detector such as detecting a chair next to a table and missing the same chair in a football field. For some applications, the context should not matter like an object placed in a box of a retail system. For other applications, the context is an important cue, such as images from an MRI scanner. In this paper we evaluate the effect of context on popular deep object detectors.

For deep object detectors, there are works on the context around the object [1], as scene context [2–4], as combinations of local parts and the global image structure [5, 6], by using multi-scale fusion and attention [7], and by using recurrent neural networks [8]. Besides, to have a robust decision, [9] uses mixup augmentation and [10] investigates how to disentangle object from its co-occurring context. [11] investigates the contextual effect on visual recognition with various ways and compare with human performance. [12] explores the bounds of object classes by using contextual information and show the cases when is beneficial to use or discard. Differently, we classify object detectors in terms of context use and analyze the effect of context for these detector types.

Popular deep object detectors follow either a single-stage or a two-stage approach. Two-stage detectors consist of class agnostic region proposals and detection parts. RCNN [13] is the earliest deep two-stage detector that crops the ROIs from the input image before feeding the CNN backbone, without accessing context. Faster RCNN [14] introduces the trainable Region Proposal Network (RPN) and each candidate region is cropped from deep featuremap. Faster RCNN is the most common two-stage detector and a great inspiration to other detectors [5, 6, 15, 16], hence we evaluate Faster-RCNN as the prototypical two-stage detector. Single-stage detectors do not use any proposal method and obtain the detection in a single run. YOLO [17] treats detection as a regression problem and detects the objects from a full image. In YOLOv2 [18] and YOLOv3 [19], the method is improved by using a deeper backbone model, multi-scale training, high resolution input and anchor boxes. SSD [20] predicts category scores and box offsets for a fixed set of anchors from different scales. RetinaNet [21] proposes focal loss which focuses on the hard training samples and converges faster. EfficientDet [22] scales the model and proposes fast multi-scale feature fusion. For our experiments, we choose YOLOv3 since it uses anchor boxes and multi-scale training, thus comparable with Faster RCNN.

In this paper, we assume that context is formed as everything around the object including other objects. We classify object detectors on how they use context (Fig. 3.1): (i) no context (RCNN), (ii) partial context (Faster RCNN) and (iii) full context (YOLO) on a

fully-controlled dataset, see Fig. 3.2. Using context is beneficial if the object correlates with its environment. However, the performance is reduced when the context is incoherent. We demonstrate the effect of context on detector performance by increasing the context around to object in each testing case. Also, we indicate how much contribution the detector can obtain by only using the context information without a visible main object.

We have the following contributions: First, we show that modern deep object detectors can access context by virtue of their receptive field size even if the object regions are cropped from the featuremap. Second, the effect of context is evaluated quantitatively on most common object detection networks. To conclude, we indicate that single and two stages networks employ contextual information except methods crop the ROIs from the input such as RCNN.

3.2. EXPERIMENTAL EVALUATION OF CONTEXT

We analyze the effects of various contextual correlations for common detector types on a fully-controlled context dataset and evaluate context with natural images.

We categorize deep object detectors (see Fig. 3.1) as:

Crop-Input. We base this class of detectors on the seminal RCNN [13] approach, which originally uses class agnostic object proposal bounding boxes [23] that are cropped from the input and fed to a CNN backbone for feature extraction, then the extracted features are used for detection. Since the method crops the proposals from the input image before feature extraction, it does not access any context beyond the bounding box, thus we call it 'no context' method. In reality, a network may retrieve minimal context between an object and the area inside the bounding box.

Crop-FM. This class of detectors crops bounding boxes from CNN featuremaps. The seminal example is based on Faster RCNN [14] which has two stages: a detection head for object classification and an RPN which outputs candidate object boxes. These boxes are cropped by ROI pooling from featuremaps. These featuremaps are deep in the network and thus are the result of convolutions with a large receptive field. Due to such large receptive fields, the featuremap crops include context information beyond the cropped regions which can be exploited for recognition.

NoCrop. This class of object detectors does not crop at all and includes most of the single-stage object detectors such as YOLO detectors [17–19]. Predictions are made by using the full featuremap and thus can exploit all context.

3.2.1. EVALUATING OBJECT-CONTEXT CORRELATIONS

It is difficult to vary the correlation between an object and its context in real images. Thus, we create a fully controlled context-sensitive dataset from the 10-class Fashion MNIST [24]. To vary object-context correlation we create 6,000 images (2000 per training, validating and test set) to form the Quadrant-FMNIST (Q-FMNIST) dataset by placing images in quadrants. We create a 2-class object detection problem where the top-left quadrant has the object of interest (class-1: 'Pullover' and class-2: 'Shirt') and the other 3 quadrants are filled with images of other 8 classes which is how we vary object-context correlation. Namely, these 8 classes become background for each image. We have 5

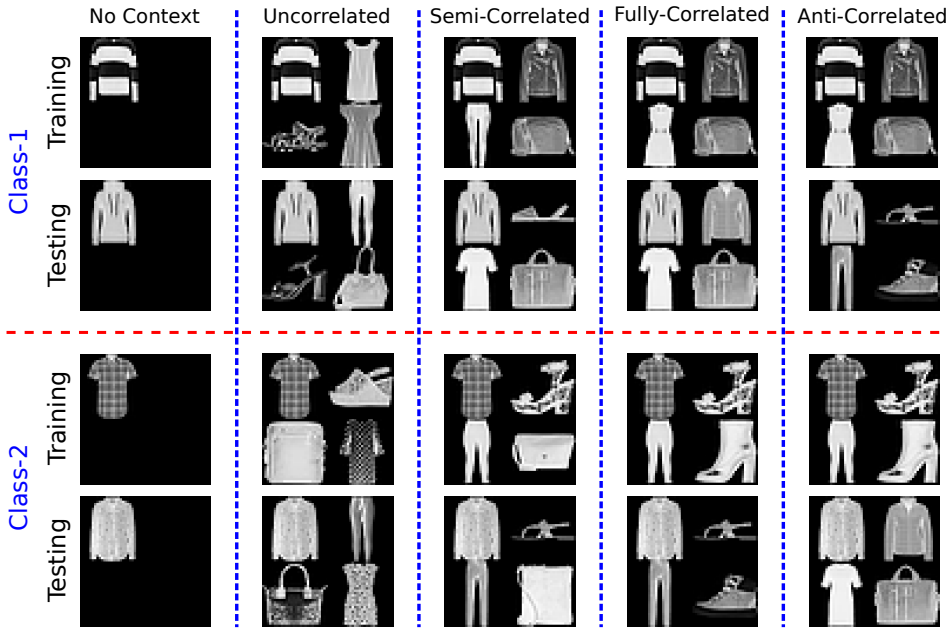


Figure 3.2: Q-FMNIST training and testing samples from each class. Controlled setting where the main object (top left) is surrounded by a varying degree of object-context correlations.

degrees of object-context correlations, shown in Fig. 3.2.

No context: The background is full black.

Uncorrelated context: The 3 locations of the template are filled randomly from the 8 context classes.

Semi-correlated context: Up-right corner, bottom-left corner and bottom-right corner locations of class-1 are filled with respectively 'Dress', 'Coat' and 'Bag' classes (common context for class-1) by the probability of 75% and with 'Trouser', 'Sandal' and 'Ankle bot' classes by 25%. Similarly, class-2 images are filled with 'Trouser', 'Sandal' and 'Ankle bot' classes (common context for class-2) by 75% and 'Dress', 'Coat' and 'Bag' classes by 25%.

Fully-correlated context: This mode represents the cases when the objects are observed always in similar context, such as cows and horses are together on a grass field. The context is placed in a structured way in terms of location, the same context occurs in the same place in the train and test set. For class-1 images, coat, dress and bag objects are placed respectively up-right corner, bottom-left corner and bottom-right corner. Likewise, For class-2, sandal, trouser and ankle bot objects are placed with the same order as the coat, dress and bag objects. The context objects are not alternated.

Anti-correlated context: Training set is built by using fully-structured context train set, however, testing is done by filling with incoherent context by switching the class-specific context. This mode illustrates the cases when the object is seen in an unusual context, such as a shoe in a plate. Class-1 images are filled with 'Trouser', 'Sandal' and

Context	Crop-Input	Crop-FM	NoCrop
No context	87.9 ± 1.0	$\mathbf{88.1 \pm 0.4}$	86.9 ± 0.7
Uncorrelated	$\mathbf{87.9 \pm 1.0}$	86.7 ± 0.7	83.7 ± 1.1
Semi-correlated	87.9 ± 1.0	89.3 ± 0.8	$\mathbf{90.4 \pm 0.5}$
Fully-correlated	87.9 ± 1.0	99.7 ± 0.1	$\mathbf{100 \pm 0.0}$
Anti-correlated	$\mathbf{87.9 \pm 1.0}$	1.8 ± 0.2	0.0 ± 0.0

Table 3.1: Accuracy on Q-FMNIST. Context affects Crop-FM and NoCrop detectors. For correlated context results improve. Uncorrelated context is worse than no context. Anti-correlated context is detrimental.

'Ankle bot' classes and likewise class-2 images are filled with 'Dress', 'Coat' and 'Bag' classes respectively up-right corner, bottom-left corner and bottom-right corner. Namely, the context of class-1 and class-2 are swapped.

We instantiate each of the three deep object detector classes in Fig. 3.1 with two convolution layers with 6 and 16 3x3 filters, two max pooling layers, one fully connected layer with 128 neurons and softmax classifier. We use the ground truth location to crop bounding boxes for the Crop-Input model. For the Crop-FM model, RoiAlign [25] is used for cropping ROIs from the featuremap. Each method is trained 5 times for 15 epochs with the AdaDelta optimizer.

Results. In Table 4.1, the Crop-Input detector disregards all context. For the Crop-FM and NoCrop detectors, the 'no context' setting gives an object-only baseline. Adding 'semi-structured context' improves, and adding 'fully correlated context' even more. Interestingly, adding 'uncorrelated context' decreases results, whereas 'anti-correlated context' completely misclassifies the objects. As the NoCrop detector uses the full image, it is more sensitive to context changes than the Crop-FM. Being more sensitive can be an advantage for correlated object-contexts, yet can be detrimental for random context or when an object is placed outside the usual context where the context may outweigh the object itself.

3.2.2. EVALUATING CONTEXT ON NATURAL IMAGES

We investigate the context for natural images on the COCO minival 2014 split [26]. We evaluate two settings: *Hiding background* and *Hiding foreground*, see Fig. 3.3. We begin with the ground truth object bounding box. For *hiding BG*, we start without any context using a black background and incrementally add more background pixels on each side of the object. For *hiding FG*, we start without an object and make the bounding box black and incrementally add object pixels towards the center on each side. We increase pixels in the range $\in \{0, 5, 10, 25, 50, 100, 150, 200, 250, ..\}$ until reaching the full image for both settings. If the context of an image reached an image border on one side, it stops there, yet, the change continues on the other object sides.

For Crop-Input we use a variation of R-CNN [13] where we use a softmax classifier based on an ImageNet pretrained Alexnet. The object crops are resized as 227x227 and trained from scratch 35 epochs with SGD for an initial learning rate of 1e-3. For Crop-FM, we use Faster RCNN with ROI Align to crop ROIs from the featuremap. An existing COCO pretrained network is used with a Resnet-50 backbone with an FPN [15]. For NoCrop, we

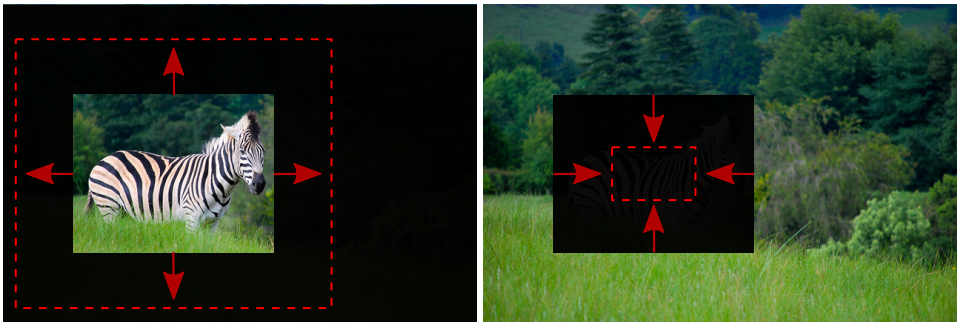


Figure 3.3: Example of hiding background (left) and hiding foreground (right). Hiding BG incrementally adds background pixels. Hiding FG incrementally adds object pixels.

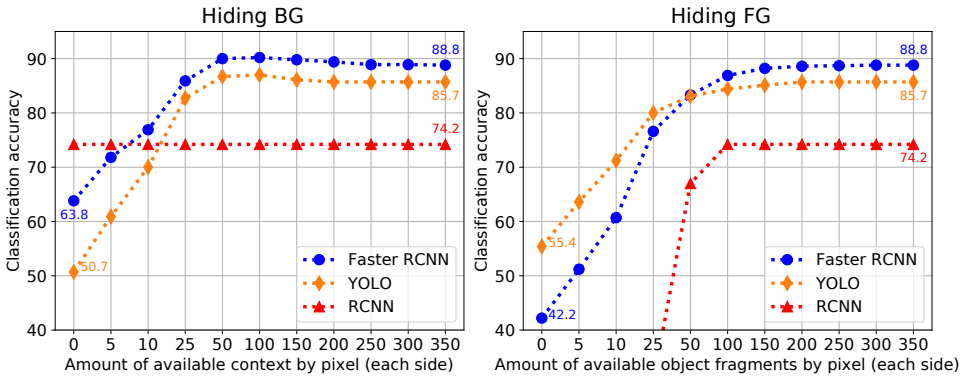


Figure 3.4: Hiding Background and Foreground experiments on 80-class COCO with RCNN, Faster RCNN and YOLO. The x-axis shows how many pixels of context (Hiding BG) or object (Hiding FG) is available from each side. Both plots confirm Faster RCNN is less sensitive to context than YOLO. For Hiding BG, 50-100 pixels extra has best object-context correlation, which reduces when adding more background. For Hiding FG, R-CNN needs only 100 pixels to classify an object.

use YOLO version 3 [19] which is fully-convolutional and has 75 layers with skip connections. To evaluate hiding BG and FG setups, we use classification accuracy. The effect of localization is minimized as following: For RCNN, ground truth box locations are used to crop the objects from the input image. For Faster RCNN, IoU threshold is set as 0.25. For YOLO, the prediction is counted as correct If the center location of predicted and ground truth boxes for correct class label remain in the same grid cell.

Hiding BG. Results in Fig. 3.4 (left) confirms that the Crop-Input R-CNN does not depend on context. The Crop-FM Faster RCNN outperforms the NoCrop YOLO when no context is available. Both detectors have their peak when 100 pixels is added to each side of the object: 90.2% for Faster RCNN and 87% for YOLO. We hypothesize that the object-context correlation is best around 100 pixels and decreases as more background is taken into account.

For *Hiding BG*, in Fig. 3.5 (top) we show the classes that have the smallest and the largest difference between no context and full context. For Faster RCNN, *bottle* class is

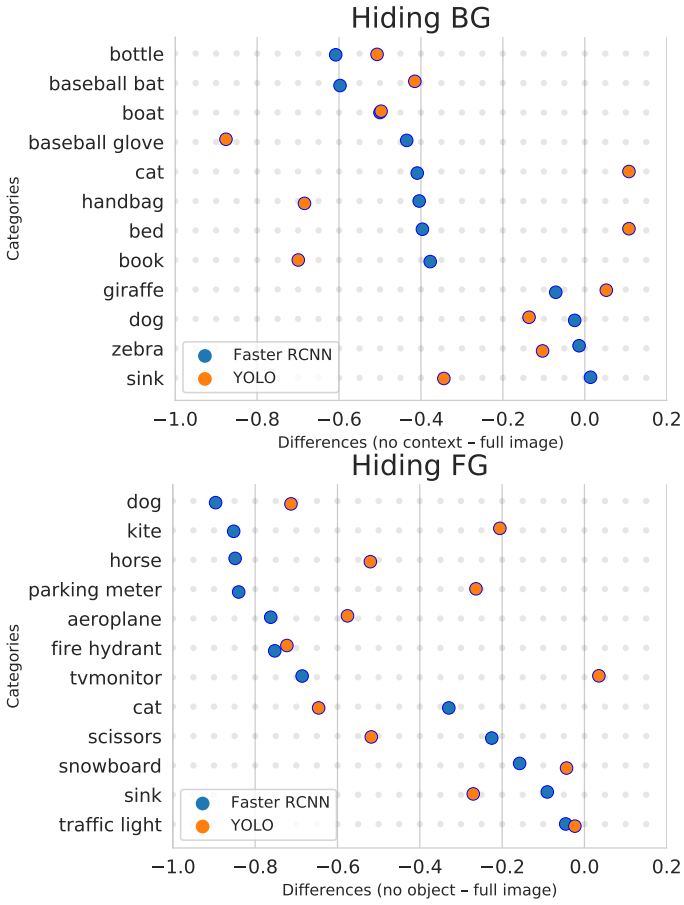


Figure 3.5: Evaluation of class-specific result of Faster RCNN and YOLO on COCO, (top) Hiding Background and (bottom) Hiding Foreground. Data points represent the 3 best and 3 worst-performing classes for each method according to difference between context/object and full image.

highly context dependent and loses 61% without context. Besides, *sink* performs 2% better without context. For YOLO, *baseball glove*, *handbag* and *book* lose more than 65% due to lack of context. However, *cat*, *bed* and *giraffe* obtain better result when no context is used. Interestingly, 3/4 of classes obtains better result with using some amount of context rather than using full context. The fact also explains the performance increases in Fig. 3.4 (left) between 50 and 300 context ranges.

Hiding FG. Results in Fig. 3.4 (right) shows that YOLO is best in exploiting context when the full object is removed. Interestingly, the performance of Faster RCNN is still far from random for 80 classes without actually seeing the object while RCNN scores truly random with $\frac{1}{80} \approx 1.3\%$. RCNN can classify an object when more pixels are available and after 100 pixels, adding more pixels does not help. Surprisingly, when comparing *Hiding*

FG with *Hiding BG* it shows that YOLO is 4.7% better for not having the object when compared to not having the context.

For *Hiding FG*, in Fig. 3.5 (bottom) we show the classes that have the smallest and the largest difference between no object and full image. Without seeing the actual object, Faster RCNN and YOLO can still classify a *traffic light*. Classes like *dog* and *fire hydrant* lose more than 70% performance for both methods when no object parts are visible. These classes have high robustness to context change (Fig. 3.5 - top), thus their object parts are crucial for their detection. Surprisingly, *tvmonitor* can be identified by YOLO 3.5% better without seeing the object itself.

3.3. DISCUSSION AND CONCLUSION

In this paper, we investigate the effect of context on 3 different deep object detectors, (i) cropping the input (RCNN), (ii) partial context (Faster RCNN) and (iii) full context (YOLO). Experiments with Q-FMNIST and COCO datasets show that single and two stage methods access the context because of their large receptive fields excluding RCNN since it crops the ROIs from the input. Hiding BG and FG experiments indicate that context often improves the result until some extend and sometimes it degrades the performance. For YOLO, having no object visible outperforms having no context visible.

Generating realistic toy dataset for context experiments is challenging. Even if Q-FMNIST dataset is limited, it still provides controlled-context setup to compare common detectors. Besides, in hiding BG and FG experiments, object size matters and the effect of object size may supply insightful results, however, we focus on overall and class-specific performance indications rather than object sizes.

REFERENCES

[1] S. Gidaris and N. Komodakis, *Object detection via a multi-region and semantic segmentation-aware cnn model,* in *ICCV* (2015).

[2] S. Gupta, B. Hariharan, and J. Malik, *Exploring person context and local scene context for object detection,* arXiv preprint arXiv:1511.08177 (2015).

[3] J. Sun and D. W. Jacobs, *Seeing what is not there: Learning context to determine where objects are missing,* in *CVPR* (2017).

[4] Y. Liu, R. Wang, S. Shan, and X. Chen, *Structure inference net: Object detection using scene-level context and instance-level relationships,* CVPR (2018), 10.1109/cvpr.2018.00730.

[5] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, and H. Lu, *Couplenet: Coupling global structure with local parts for object detection,* in *ICCV* (2017).

[6] X. Fan, H. Guo, K. Zheng, W. Feng, and S. Wang, *Object detection with mask-based feature encoding,* arXiv preprint arXiv:1802.03934 (2018).

[7] J.-S. Lim, M. Astrid, H.-J. Yoon, and S.-I. Lee, *Small object detection using context and attention,* arXiv preprint arXiv:1912.06319 (2019).

[8] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, *Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks,* in *CVPR* (2016).

[9] Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, and M. Li, *Bag of freebies for training object detection neural networks,* arXiv preprint arXiv:1902.04103 (2019).

[10] K. K. Singh, D. Mahajan, K. Grauman, Y. J. Lee, M. Feiszli, and D. Ghadiyaram, *Don't judge an object by its context: Learning to overcome contextual bias,* in *CVPR* (2020).

[11] M. Zhang, C. Tseng, and G. Kreiman, *Putting visual object recognition in context,* arXiv preprint arXiv:1911.07349 (2019).

[12] E. Barnea and O. Ben-Shahar, *Exploring the bounds of the utility of context for object detection,* in *CVPR* (2019).

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Region-based convolutional networks for accurate object detection and segmentation,* PAMI (2015).

[14] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks,* in *Advances in neural information processing systems* (2015) pp. 91–99.

[15] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, *Feature pyramid networks for object detection,* in *CVPR* (2017).

[16] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, *Deformable convolutional networks,* in *ICCV* (2017).

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection,* (2016), arXiv:1506.02640 [cs.CV] .

[18] J. Redmon and A. Farhadi, *Yolo9000: Better, faster, stronger,* in *CVPR* (2017).

[19] J. Redmon and A. Farhadi, *Yolov3: An incremental improvement,* arXiv preprint arXiv:1804.02767 (2018).

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, *Ssd: Single shot multibox detector,* in *ECCV* (2016).

[21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, *Focal loss for dense object detection,* in *ICCV* (2017).

[22] M. Tan, R. Pang, and Q. V. Le, *Efficientdet: Scalable and efficient object detection,* arXiv preprint arXiv:1911.09070 (2019).

[23] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, *Selective search for object recognition,* International journal of computer vision **104**, 154 (2013).

[24] H. Xiao, K. Rasul, and R. Vollgraf, *Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,* arXiv preprint arXiv:1708.07747 (2017).

[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, *Mask r-cnn,* in *ICCV* (2017).

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft coco: Common objects in context,* in *ECCV* (2014).

4

Hallucination in Object Detection - A Study in Visual Part Verification

Repetition does not transform a lie into a truth.

Franklin D. Roosevelt

This chapter has been published as:
O. S. Kayhan, B. Vredebregt, and J. C. van Gemert, Hallucination in object detection — a study in visual part verification, in 2021 IEEE International Conference on Image Processing (ICIP) (2021) pp. 2234–2238. [1]

ABSTRACT

We show that object detectors can hallucinate and detect missing objects; potentially even accurately localized at their expected, but non-existing, position. This is particularly problematic for applications that rely on visual part verification*: detecting if an object part is present or absent. We show how popular object detectors hallucinate objects in a visual part verification task and introduce the first visual part verification dataset: DelftBikes*[1]*, which has 10,000 bike photographs, with 22 densely annotated parts per image, where some parts may be missing. We explicitly annotated an extra object state label for each part to reflect if a part is missing or intact. We propose to evaluate visual part verification by relying on recall and compare popular object detectors on DelftBikes.*

4

[1] https://github.com/oskyhn/DelftBikes

4.1. INTRODUCTION

Figure 4.1: Hallucination examples on DelftBikes for Faster RCNN [2], RetinaNet [3] and YOLOv3 [4]. Faster RCNN and RetinaNet detect the front wheel and YOLOv3 predicts the saddle with a high IoU score. Deep object detectors may detect non-existent objects at their expected locations.

Automatically localizing and detecting an object in an image is one of the most important applications of computer vision. It is therefore paramount to be aware that deep object detectors can hallucinate non-existent objects, and they may even detect those missing objects at their expected location in the image, see Fig. 4.1. Detecting non-existing objects is particularly detrimental to applications of automatic *visual part verification* or *visual verification*: determining the presence or absence of an object. Examples of visual verification include infrastructure verification in map making, missing instrument detection after surgery, part inspections in machine manufacturing etc. This paper shows how popular deep detectors hallucinate objects in a case study on a novel, specifically created visual object part verification dataset: DelftBikes.

Visual verification as automatic visual inspection is typically used for manufacturing systems with applications such as checking pharmaceutical blister package [5], components on PCBs [6, 7], solder joint [8], parts of railway tracks [9], rail bolts [10], aeronautic components [11, 12], objects [13], and parts under motion [14]. In this paper, we do not focus on a particular application. Instead, we evaluate generic deep object detectors which potentially can be used in several visual inspection applications.

There are important differences between visual verification and object detection. An object detector should not detect the same object multiple times. For visual verification, however, the goal is to determine if an object is present or absent, and thus having an existing object detected multiple times is not a problem, as long as the object is detected at least once. This makes recall more important than precision. Moreover, there are differences in how much costs a mistake has. The cost for an existing object that is

Figure 4.2: Example images of our DelftBikes visual verification dataset. Each image has a single bike with 22 bounding box annotated parts. The similar pose, orientation and position can be misleading for context-sensitive detectors as often one or two parts are missing (the saddle in (a), the wheels in (e) etc.).

not detected (false negative) is that a human needs to check the detection. The cost for a missing object that is falsely hallucinated as being present (false positive) is that this object is a wrongly judged as intact and thus may cause accidents in road infrastructure, or may cause incomplete objects to be sent to a customer. The costs for hallucinating missing objects is higher than missing an existing object. These aspects motivate us to not use the evaluation measure of object detection. Object detectors are typically evaluated with mean Average Precision (mAP) and because detections of non-existent objects at lower confidence levels does not significantly impact mAP, the problem of object hallucination has largely been ignored. Here, we propose to evaluate visual verification not with precision but with a cost-weighted variant of recall.

Object hallucination by deep detectors can be causes by sensitivity to the absolute position in the image [15, 16] while also affected by scene context [17–21]. Here, we focus on the visual verification task, its evaluation measure, a novel dataset, and a comparison of popular existing detectors. Investigating context is future work.

Existing object detection datasets such as PASCAL VOC [22], MS-COCO [23], Imagenet det [24], and Open Image [25] have no annotated object parts. Pascal-Parts [26] and GoCaRD [27] include part labels, yet lack information if a part is missing and where, as is required to evaluate visual verification. Thus, we collected a novel visual verification dataset: DelftBikes where we explicitly annotate all part locations and part states as missing, intact, damaged, or occluded.

We have the following contributions:

1. We demonstrate hallucination in object detection for 3 popular object detectors.

2. A dataset of 10k images with 22 densely annotated parts specifically collected and labeled for visual verification.

3. An evaluation criteria for visual verification.

4.2. DELFTBIKES VISUAL VERIFICATION DATASET

DelftBikes (See Fig. 4.2) has 10,000 bike images annotated with bounding box locations of 22 different parts where each part is in one of four possible states:

intact: The part is clearly evident and does not indicate any sign of damage. All the images in Fig. 4.2 have an intact steer.

damaged: The part is broken or has some missing parts. In Fig. 4.2-g, the front part of the saddle is damaged.
absent: The part is entirely missing and is not occluded. Fig. 4.2-e has missing front and back wheels.
occluded: The part is partially occluded because of an external object or completely invisible. The saddle in Fig. 4.2-b is covered with a plastic bag.

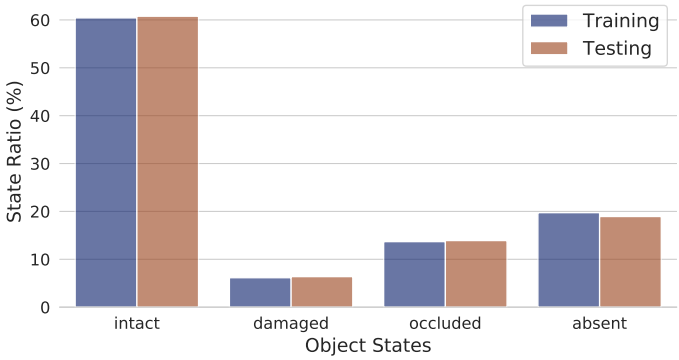


Figure 4.3: The distribution of part states for train and test sets in DelftBikes. The ratio of part states are roughly similar for train and test sets. The intact parts have the highest ratio by around 60%. Approximately 20% of parts in the dataset are absent. The damaged and occluded parts constitute 20%.

The distribution of part states is approximately similar for training and testing set, see Fig. 4.3. The part state distribution shows 60.5% intact, 19.5% absent, 14% occluded, and 6% damaged. The *front pedal*, *dress guard*, *chain* and *back light* have respectively the highest number of intact, absent, occluded and damaged part states. Note that even if a part is absent or occluded, we still annotate its most likely bounding box location. DelftBikes contains positional and contextual biases. In Fig. 4.4 where we plot an ellipse for each part in the dataset in terms of their mean position, height and width. It is possible to recognize the shape of a bike, which indicates that there are strong part-to-part position and contextual relations. Its those biases that learning systems may falsely exploit and cause detector hallucinations.

4.3. EXPERIMENTS ON DELFTBIKES

The dataset is randomly split in 8k for training and 2k for testing. We use a COCO pre-trained models of Faster RCNN [2] and RetinaNet [3]. Both networks have a Resnet-50 [28] backbone architecture with FPN. The networks are finetuned with DelftBikes for 10 epochs using SGD with a initial learning rate of 0.005. The YOLOv3 [4] architecture is trained from scratch for 200 epochs using SGD with an initial learning rate of 0.01. Other hyperparameters are set to their defaults. We group the four part states in two categories for visual verification: (i) *missing* parts consist of absent and occluded states and (ii) *present* parts include intact and damaged states. During training, only parts with *present* states are used.

Detection. We first evaluate traditional object detection using AP. For object detec-

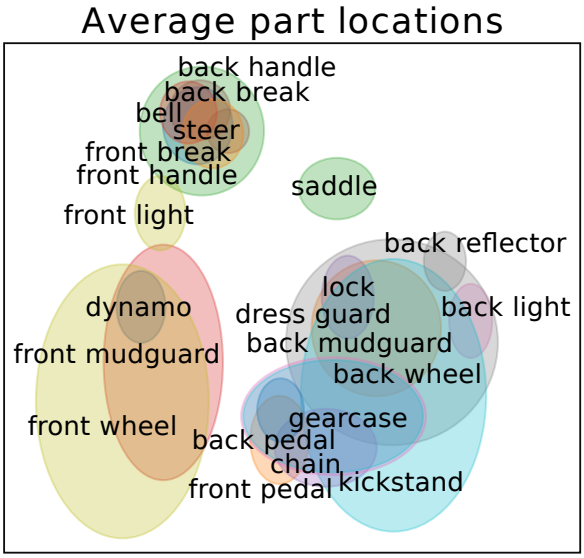


Figure 4.4: Averaging position and size for all 22 parts in DelftBikes resembles a bicycle, illustrating the prior in absolute position and the contextual part relations.

tion, the *missing* parts are not used during training nor testing. In Fig. 4.5, we show results for an IoU of 0.5:0.95 for the 3 detectors. For most of the classes, Faster RCNN and RetinaNet obtain approximately a similar result and YOLOv3 is a bit behind. *Front wheel* and *back wheel* are large and well detected. The small parts like *bell* and *dynamo* have under 12% AP score because they are small parts and often not present. The other parts are below 50% AP, where half of the parts have less than 20% AP, which makes DelftBikes already a challenging and thus interesting object detection dataset.

Recall of *missing* parts. Here, we analyze the hallucination failure of the detectors by evaluating how many non-existing parts they detect in an image. We calculate the IoU score for each detected *missing* part on the test set. We threshold these false detections in terms of their IoU scores to evaluate if the missing parts are still approximately localized. We define the recall score which is the ratio between the number of detected missing part at a given IoU threshold and the total number of missing parts.We show recall for varying IoU threshold for each method in Fig. 4.6. For a reasonable IoU of 0.5, RetinaNet and YOLOv3 detect approximately 20% of missing parts and Faster RCNN 14%. Without looking at position, (IoU=0), RetinaNet and YOLOv3 detect as much as almost 80% of *missing* parts. Interestingly, Faster RCNN, with similar mAP object detection score as RetinaNet, detects only 32% of missing parts. For Faster RCNN, the most hallucinated part with 14% is *gear case*. For YOLOv3, a missing *dynamo* is most detected and RetinaNet hallucinates most about the *dress guard*.

Evaluating visual verification. For visual verification, we want high recall of *present* parts and low recall of *missing* parts where detecting the same object multiple times does not matter. Besides, wrongly detected *missing* parts (false positives) cost more than not detected *present* parts (false negatives). Thus, our F_{vv} evaluation score is based on recall

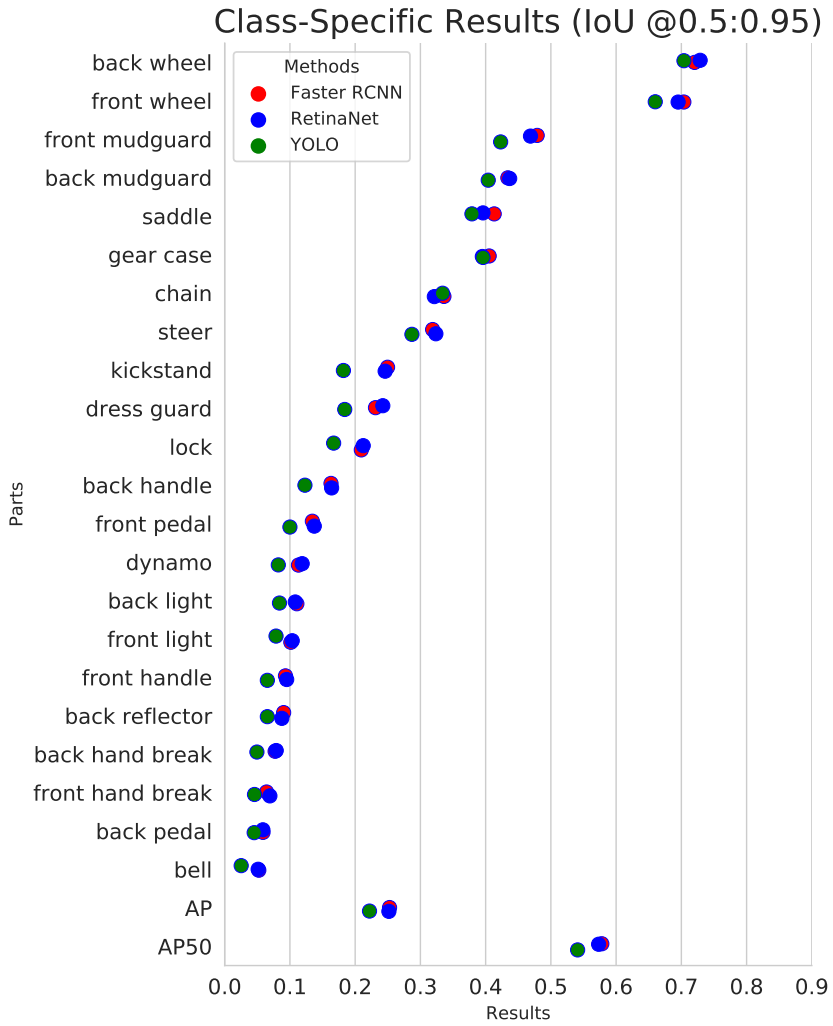


Figure 4.5: Object detection results on DelftBikes. Results per category and overall performance. Notice that half of the detections are below 20% AP score. In most of the cases, Faster RCNN and RetinaNet perform similarly and YOLOv3 is behind them.

and inspired by the F_β score [29] so we can weight detection mistakes differently as

$$F_{vv} = \frac{(1+\beta^2)R^P(1-R^M)}{\beta^2(1-R^M)+R^P}. \tag{4.1}$$

R^P is the *present* recall and R^M the *missing* recall calculated at a certain IoU threshold. The β parameter allows to weight the detection mistakes, where we set the β parameter to 0.1 so that detections of *missing* parts are 10x more costly than not detected *present* parts.

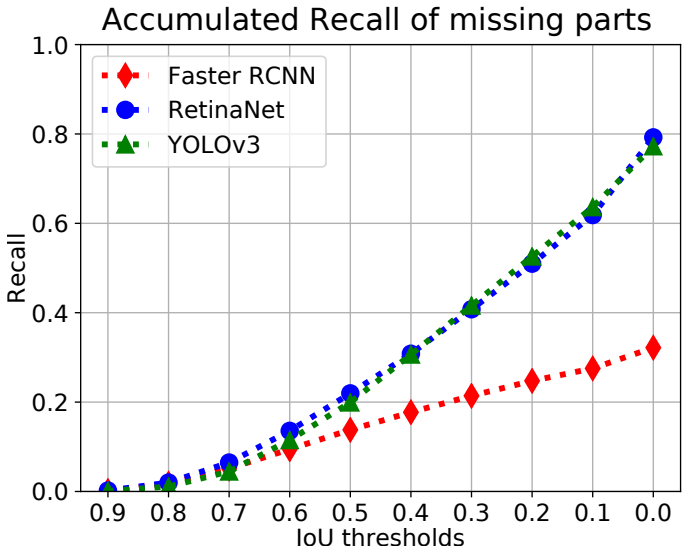


Figure 4.6: Recall of *missing* parts on DelftBikes for varying Intersection over Union (IoU). We annotated likely position of missing parts, and the recall of such missing parts should be as low as possible. All methods wrongly detect missing parts at approximately their expected location, as in Fig. 4.1.

Visual verification results. Visual verification performance is estimated by using the recall of present and missing parts. We have two setups for visual verification calculation: with and without localization. *Visual verification with localization:* the *present* recall has an IoU threshold of 0.5, where the *missing* recall is less relying on position and we set its IoU threshold to 0.1. *Visual verification without localization:* we set all IoU thresholds to 0. This, in addition, allows us to evaluate a full-image multi-class multi-label classification (MCML) approach. An Imagenet pretrained ResNet-50 architecture is fine-tuned with BCE with logits loss and SGD with an initial learning rate of 0.05 for 15 epochs. After every 5 epoch, the learning rate is reduced by a factor of 10. The network obtains 91% of recall for present parts and 32% of recall for missing parts.

Results are shown in Table 4.1. For the *with localization* results, Faster RCNN outperforms RetinaNet and YOLO in terms of lower recall of *missing* parts by 28% and a higher F_{vv} score by 72%. RetinaNet and YOLOv3 detects more than 60% of *missing* parts and achieve only 38% and 36% of F_{vv} score respectively. In Fig. 4.5, the AP scores of Faster RCNN and RetinaNet are quite similar, yet the F_{vv} performance of Faster RCNN is almost 2 times higher than RetinaNet. RetinaNet has 7% more intact recall score than YOLOv3, however, the difference for F_{vv} is only 2%. For the *without localization* results, when the *present* and *missing* IoU thresholds are set to 0, all the methods obtain more than 90% *present* recall. Interestingly, the MCML method, which only needs full image class labels, outperforms RetinaNet and YOLOv3 detectors and performs similar to Faster RCNN.

Method	T^P	T^M	R^P	R^M	F_{vv}
With localization					
Faster RCNN	0.5	0.1	0.83	**0.28**	**0.72**
RetinaNet	0.5	0.1	**0.90**	0.62	0.38
YOLOv3	0.5	0.1	0.83	0.64	0.36
Without localization					
Faster RCNN	0.0	0.0	0.92	**0.32**	**0.68**
RetinaNet	0.0	0.0	**0.99**	0.79	0.21
YOLOv3	0.0	0.0	0.95	0.77	0.23
MCML	0.0	0.0	0.91	0.32	0.68

Table 4.1: Visual verification of Faster RCNN, RetinaNet, YOLOv3 and MCML for different present (T^P) and missing (T^M) IoU thresholds on DelftBikes. (top) When (T^P, T^M) equals to (0.5, 0.1): RetinaNet has highest recall for *present* parts. Faster RCNN detects the fewest missing parts and has best F_{vv} score. (bottom) When localization is discarded: MCML method outperforms RetinaNet and YOLOv3 and results similarly Faster RCNN in F_{vv} score.

4.4. DISCUSSION AND CONCLUSION

We show hallucinating object detectors: Detectors can detect objects that are not in the image even with a high IoU score. We show hallucination in the context of a visual part verification task. We introduce DelftBikes, a novel visual verification dataset, with object class, bounding box and state labels. We evaluate visual verification by recall, where the cost of falsely detected missing parts is more expensive than a missing present part. For object detection, Faster RCNN and RetinaNet has similar AP score, however, Faster RCNN is the better for visual verification.

REFERENCES

[1] O. S. Kayhan, B. Vredebregt, and J. C. van Gemert, *Hallucination in object detection — a study in visual part verification,* in *2021 IEEE International Conference on Image Processing (ICIP)* (2021) pp. 2234–2238.

[2] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks,* in *NIPS* (2015).

[3] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, *Focal loss for dense object detection,* in *ICCV* (2017).

[4] A. Farhadi and J. Redmon, *Yolov3: An incremental improvement,* CVPR (2018).

[5] R. G. Rosandich, *Automated visual inspection systems,* in *Intelligent Visual Inspection* (1997).

[6] H. C. Garcia and J. R. Villalobos, *Automated refinement of automated visual inspection algorithms,* IEEE T-ASE (2009).

[7] D. Koniar, L. Hargas, A. Simonova, M. Hrianka, and Z. Loncova, *Virtual instrumentation for visual inspection in mechatronic applications,* Procedia Engineering (2014).

[8] T.-H. Kim, T.-H. Cho, Y. S. Moon, and S. H. Park, *Visual inspection system for the classification of solder joints,* Pattern Recognition (1999).

[9] E. Resendiz, J. M. Hart, and N. Ahuja, *Automated visual inspection of railroad tracks,* IEEE transactions on ITS (2013).

[10] F. Marino, A. Distante, P. L. Mazzeo, and E. Stella, *A real-time visual inspection system for railway maintenance: automatic hexagonal-headed bolts detection,* IEEE Transactions on Systems, Man, and Cybernetics (2007).

[11] H. Ben Abdallah, I. Jovančević, J.-J. Orteu, and L. Brèthes, *Automatic inspection of aeronautical mechanical assemblies by matching the 3d cad model and real 2d images,* Journal of Imaging (2019).

[12] M. San Biagio, C. Beltran-Gonzalez, S. Giunta, A. Del Bue, and V. Murino, *Automatic inspection of aeronautic components,* Machine Vision and Applications (2017).

[13] A.-J. Baerveldt, *A vision system for object verification and localization based on local features,* Robotics and Autonomous Systems (2001).

[14] S. Sim, P. S. Chua, M. Tay, and Y. Gao, *Recognition of features of parts subjected to motion using artmap incorporated in a flexible vibratory bowl feeder system,* AI EDAM (2006).

[15] M. Manfredi and Y. Wang, *Shift equivariance in object detection,* in *ECCV workshop* (2020).

[16] O. Kayhan and J. v. Gemert, *On translation invariance in CNNs: Convolutional layers can exploit absolute spatial location,* in *CVPR* (2020).

[17] E. Barnea and O. Ben-Shahar, *Exploring the bounds of the utility of context for object detection,* in *CVPR* (2019).

[18] S. Gidaris and N. Komodakis, *Object detection via a multi-region and semantic segmentation-aware cnn model,* in *ICCV* (2015).

[19] Y. Liu, R. Wang, S. Shan, and X. Chen, *Structure inference net: Object detection using scene-level context and instance-level relationships,* CVPR (2018), 10.1109/cvpr.2018.00730.

[20] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, and H. Lu, *Couplenet: Coupling global structure with local parts for object detection,* in *ICCV* (2017).

[21] K. K. Singh, D. Mahajan, K. Grauman, Y. J. Lee, M. Feiszli, and D. Ghadiyaram, *Don't judge an object by its context: Learning to overcome contextual bias,* arXiv:2001.03152 (2020).

[22] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, *The PASCAL VOC2012,* .

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft coco: Common objects in context,* in *ECCV* (2014).

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, *Imagenet large scale visual recognition challenge,* IJCV (2015).

[25] A. Kuznetsova, H. Rom, N. Alldrin, J, J. R. R. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, and V. Ferrari, *The open images dataset V4: unified image classification, object detection, and visual relationship detection at scale,* CoRR **abs/1811.00982** (2018), arXiv:1811.00982 .

[26] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille, *Detect what you can: Detecting and representing objects using holistic models and body parts,* in *CVPR* (2014).

[27] L. Stappen, X. Du, *et al.*, *Go-card – generic, optical car part recognition and detection: Collection, insights, and applications,* (2020), arXiv:2006.08521 [cs.CV] .

[28] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition,* in *CVPR* (2016).

[29] N. Chinchor, *Muc-4 evaluation metrics,* in *MUC4* (Association for Computational Linguistics, 1992).

5

TILTING AT WINDMILLS: DATA AUGMENTATION FOR DEEP POSE ESTIMATION DOES NOT HELP WITH OCCLUSIONS

This chapter has been published as:
R. Pytel, O. S. Kayhan, and J. C. van Gemert, Tilting at windmills: Data augmentation for deep pose estimation does not help with occlusions, in 2020 25th International Conference on Pattern Recognition (ICPR) (2021) pp. 10568–10575 [1]

ABSTRACT

Occlusion degrades the performance of human pose estimation. In this paper, we introduce targeted keypoint and body part occlusion attacks. The effects of the attacks are systematically analyzed on the best performing methods. In addition, we propose occlusion specific data augmentation techniques against keypoint and part attacks. Our extensive experiments show that human pose estimation methods are not robust to occlusion and data augmentation does not solve the occlusion problems. [1]

[1] For the code:

https://github.com/rpytel1/occlusion-vs-data-augmentations

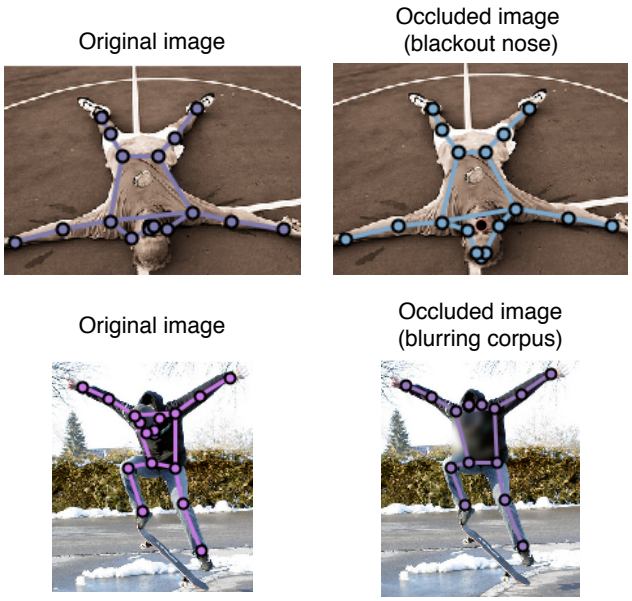


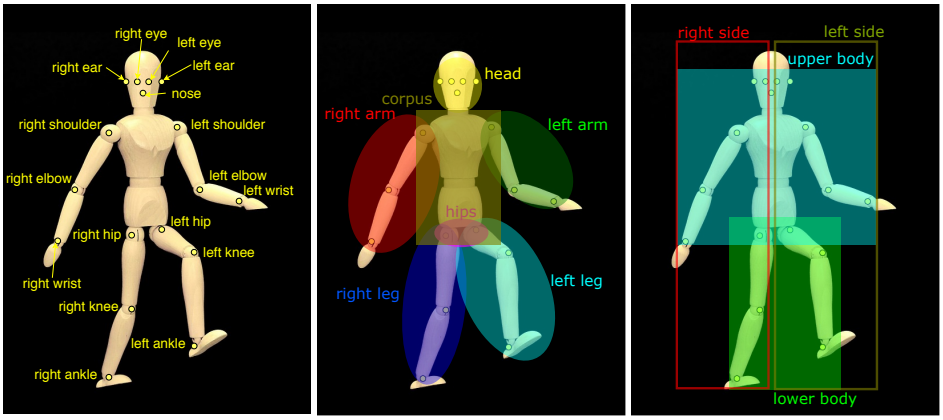
Figure 5.1: Qualitative example how HRNet-32 [19] predictions change after keypoint blackout on the nose (first row) and part blurring on the corpus (second row). For both examples keypoints change for head, nose, eyes and ears.

5.1. INTRODUCTION

Human Pose Estimation is the task of localizing anatomical keypoints such as eyes, hips, knees and localizing body-parts like head, limbs, corpus, etc., with many applications in segmentation [2–4], action recognition [5–7], pose tracking [8, 9], gait recognition [10, 11], autonomous driving [12–14], elderly monitoring [15, 16] and social behaviour analysis [17, 18]. All these applications rely on correct and robust pose estimation. In this paper we investigate the robustness of human pose estimation methods to a natural and common effect: Occlusions.

Occlusions are common and occur frequently in the wild as for example by a random object, another person [20], and self-occlusion [21]. Prior works address occlusion in a general way and exploits segmentation [13] or depth information [22]. Where [23] evaluates robustness with image and domain-agnostic universal perturbations. In contrast, we systematically analyze targeted occlusion attacks not only for keypoints, but also for and body parts and investigate the sensitivity of pose estimation to occlusion attacks.

A promising solution to occlusions is data augmentation, which is practically a default setting for deep learning applications [24] where image flipping, rotation, and scaling offer endless data variations [24–26]. As such, regional dropout and mixup methods improve the generalization performance of image classification [27–34], object localization and detection [35–37] and segmentation [38]. In pose estimation, [39] applies region based augmentation by exchanging a single keypoint patch with a random background patch. More recent approaches [19, 40] use half-body augmentation wherewith the pres-



(a) COCO keypoints annotations. (b) Part mapping for smaller parts. (c) Part mapping for larger parts.

Figure 5.2: Visualization of keypoint annotations in COCO dataset and proposed part mapping.

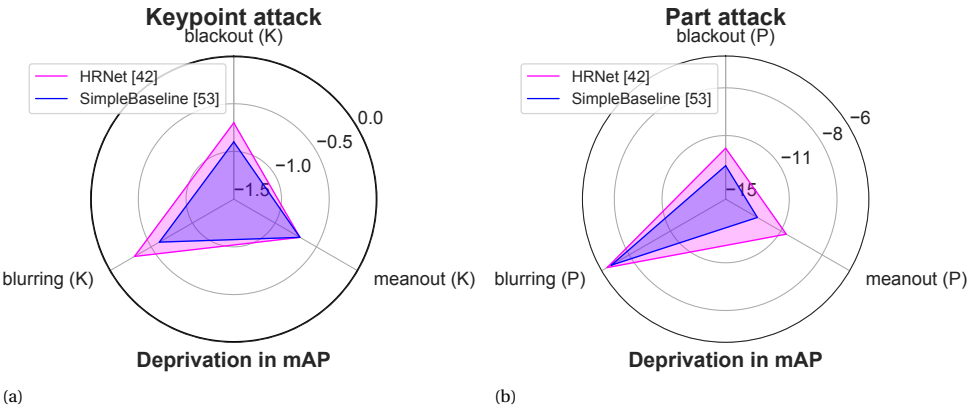
5

ence of more than 8 keypoints, by choosing upper or lower body keypoints. We implement systematic data augmentation methods for occlusion for keypoint and body parts to investigate how data augmentation can remedy occlusion attacks.

We have the following contributions: First, we conduct a structured investigation on the occlusion problem of pose estimation and introduce occlusion attacks. Second, we investigate occlusion-based data augmentation methods. Third, we show that data augmentation does not provide robustness to occlusion attacks.

5.2. RELATED WORK

Human Pose Estimation. Deep learning methods in human pose estimation can be divided into 2 categories: bottom-up and top-down. Bottom-up approaches [25, 41, 42], firstly localize identity-free keypoints and then group them into person instances. Top-down approaches [19, 40, 43, 44] firstly detect a person in the image and then perform a single person estimation within the bounding box. The top-down approaches achieve the state of the art results on various multi-person benchmarks such as COCO [45], MPII [46]. Within top-down approaches 2 categories can be distinguished: regressing direct location of each keypoint [47, 48] and keypoint heatmaps estimation [19, 40, 44, 49, 50] followed by choosing the locations with the highest heat values as the keypoints. The best performing methods on COCO keypoint challenge use a cascade network [43, 51] to improve keypoint prediction. The 'SimpleBaseline' [40] proposes simple but effective improvement by adding few deconvolutional layers to enlarge the resolution of output features. HRNet [19] which is built from multiple branches can produce high-resolution feature maps with rich semantics and performs well on COCO. Some works advance performance of HRNet via improvement over standard encoding and decoding of heatmaps [52] and basing data processing on the unit length instead of pixels [53] with an additional off-set strategy for encoding and decoding. Because of their good accuracy



(a) (b)

Figure 5.3: Robustness comparison of HRNet [19] and SimpleBaseline [40] against (a) keypoint and (b) part occlusion attacks. HRNet is more robust against both attacks, yet both attacks drop performance, where part attacks deteriorate more.

5

and wide adaptation, we focus on top-down methods, HRNet and SimpleBaseline and bottom-up approach Higher HRNet.

Occlusion in pose estimation. Occlusion in pose estimation is an under-studied problem. In [23] analyses of occlusions are done for deep pose estimators by domain-agnostic universal perturbations. More recently, attempts to solve the occlusion problem in pose estimation are suggested via the usage of segmentation of occluded parts [13] and depth of in an image [22]. OcclusionNet [54] predicts occluded keypoints via graph-neural networks yet it is applied only on vehicles. Different from these methods, in our paper we introduce keypoint occlusion attacks and body part occlusion attacks and give a structured analysis of occlusion on human pose estimation.

Data augmentation. Data augmentation is a strong, simple and popular approach to increase model robustness. Removing part of the image improves generalization of image classification [27, 32, 34] and object localization-detection [35–37]. Mixup [28, 30, 33] approaches which create a combination of two images are often used in image classification. [38][55] combine regional dropout and MixUp methods for image segmentation [38] and image classification [55] task. [39] proposes a cutmix-like approach where a small patch from the background is pasted on the single keypoint or vice versa. For the human pose estimation methods [47, 50, 56], scaling, rotation and flipping is commonly used as data augmentation. Random cropping is also used in bottom-up approaches [25, 41, 42]. More recent top-down approaches [19, 40, 43] employ the usage of half body transform by a probability of 0.3 choosing either upper or lower body keypoints. We introduce and evaluate new data augmentation methods for keypoint and for body parts specifically designed against occlusion attacks for human pose estimation.

5.3. SENSITIVITY TO OCCLUSION ATTACKS

We investigate the effect of occlusion attacks on MS COCO dataset [45]. COCO contains challenging images with the unconstrained environment, different body scales, va-

riety of human poses and occlusion patterns. The dataset contains over 200k images with 250k person instances labelled with 17 keypoints. Models are trained on COCO train2017 datasets which includes 57k images and 150k person instances. The evaluation is done on val2017 set which contains 5k images.

The occlusion attack experiments are conducted with HRNet [19] and Simple Baseline [40] for two aspects: (i) keypoint attacks, where the occlusion area is a centred circle on the chosen keypoint, (ii) body part attacks, where the occlusion area is the minimum rectangle covering all keypoints of a chosen part. The COCO keypoints and the proposed groups of body parts can be seen in Figure 5.2. For the analyses, COCO pretrained HRNet and Simple Baseline are evaluated by the performance of the network against keypoint and part occlusion attacks on COCO validation set.

HRNet and SimpleBaseline produce heatmap instead of predicting direct single location for each keypoint. The ground truth heatmaps are generated by using 2D Gaussian of size 13x13. Thus, as a default, we choose the size of the occlusion circle with a radius of 6 pixels for keypoint attacks to cover the keypoint heatmap. We have 3 different keypoint attacks: (i) Gaussian blur (blurring) attack, (ii) attack by filling with black pixels (blackout), (iii) attack by filling with a mean intensity value of a given image (meanout).

Body parts occlusion attacks are designed to draw a minimum rectangle which covers all the keypoints of a chosen part. Similar to the keypoint attacks, we have 3 different part attacks which are applied to the occlusion area: blurring with the kernel size 31 and sigma 5, blackout and meanout. These attacks can be applied on both small parts such as head, arms, hips and larger parts like upper body, lower body, left and right side (Figure 5.2 b and c).

We compare HRNet and Simple Baseline according to their robustness to keypoint and part occlusion attacks. Figure 5.3 shows that both attacks are quite successful as occlusion causes the performance to drop. HRNet is more robust against keypoint and part occlusion attacks. For further analyses, we only use HRNet as a baseline for our investigations.

5.3.1. HOW SENSITIVE TO KEY POINT OCCLUSION ATTACKS?

First, we analyze the effect of the occlusion size on the average performance of the pose estimator on all keypoints. Figure 5.4 indicates that pose estimator performance is inversely proportional to the occlusion size and blurring, blackout, and meanout attacks on average perform similarly. The size of the occlusion decreases the average performance of the estimator by approximately 3% when the radius of the occlusion circle is chosen as 18 pixels.

Second, we show the class-specific performance drops for each individual keypoints for each attack. In Figure 5.5, attacking nose causes serious loss in mAP, almost 5% for blackout, 4.4% for meanout and 1.2% for blurring. The empirical results indicate that **the nose** is the most important keypoint since the occlusion of the nose causes notable performance drop. After the nose, each eye influences the performance of other keypoints mostly by approximately 1% with each occlusion attack. Keypoints from less densely annotated places like ankles or wrists are the least influential.

If we check the analysis of the reduced accuracy per keypoint for the case of attacking nose (Figure 5.6a), the most affected keypoints are the ones within close distance,

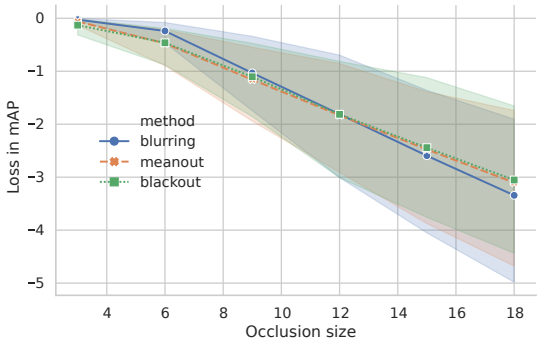


Figure 5.4: The relation between occlusion size and average loss in performance for keypoint level methods. Occlusion size greatly affects the performance.

which are eyes and ears due to being a part of the head. Interestingly, occluding nose affects the performance of the left eye estimation more than occluding the left eye itself, respectively by approximately 10% and 5% (Figure 5.6a, 5.6b). If we investigate per keypoint performance for occluding left ankle, it can be seen that the deprivation is by several magnitudes smaller than in case of the nose or left eye occlusions. From the observation of the analyses, it can be drawn that HRNet [19] is not robust to keypoint occlusion attacks.

5

5.3.2. HOW SENSITIVE TO PART OCCLUSION ATTACKS?

We analyze the effect of the part occlusion attacks on each body parts given in Figure 5.2. Attacking the upper body, left and right sides influence the overall performance the most, by more than 44%, 24% and 24% with blackout attack respectively since these three parts include the majority of the keypoints (Figure 5.7). When we examine keypoint-specific accuracy drops for the remaining keypoints of the upper body, it is clear that blackout is the most influential attack, with a drop of almost 3% for left and right ankle (Figure 5.8a). If we investigate per-keypoint behaviour for the corpus (Figure 5.8b), we observe significant degradation of the performance on all the keypoints, with left and right ankle affected the most. Interestingly, attacking on one side improves performance of the the other side (Figure 5.8c). Attacking on left side increases the mAP score of right side such as shoulder, ear, elbow keypoints. The analysis demonstrates that the pose estimator is sensitive to part occlusion attacks.

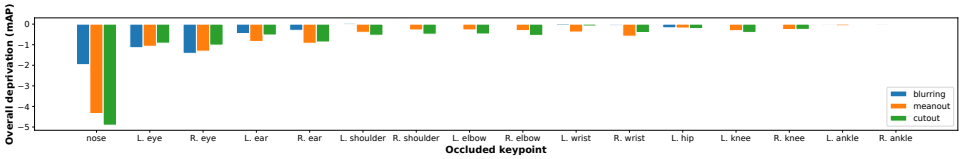
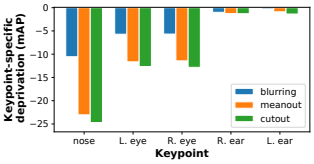
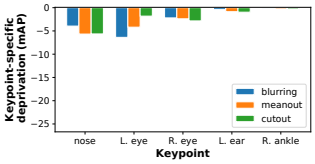


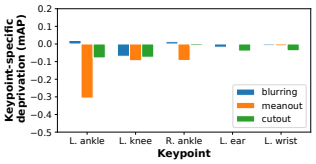
Figure 5.5: Overall loss in mAP after performing keypoint level occlusion. *L.* and *R.* correspond to the left and right side respectively. To note that, the occluded keypoint is included in the evaluation. Occluding nose causes the highest loss in performance.



(a) The nose is the most influential keypoint causes a significant drop in the performance for the closest keypoints - left eye and right eye by around 10%.



(b) When we occlude the left eye, there is a smaller loss in keypoint-specific performance for the left eye than while occluding nose.



(c) Left ankle is one of the least influential keypoints with loss only visible for meanout for occluded keypoint.

Figure 5.6: Loss in AP for top 5 keypoints with largest deprivation, when an individual key point is occluded.

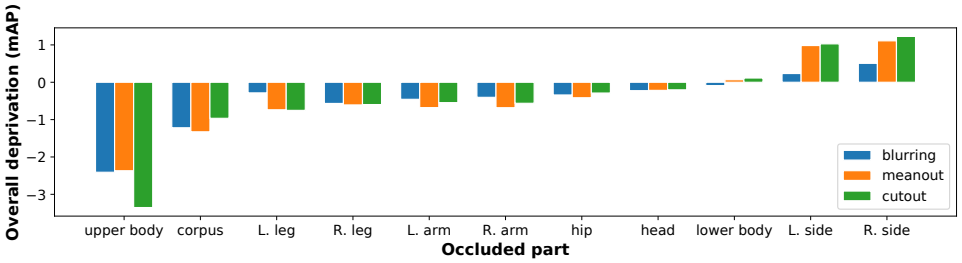
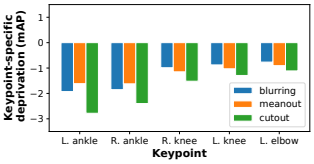
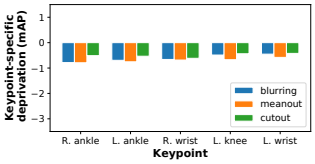


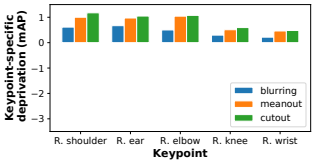
Figure 5.7: Change in mAP for various parts occluded. Upper body and sides are the parts that cause the highest loss in the performance.



(a) Significant loss in performance for all of the remaining keypoints. Blackout affects the method most.



(b) Similar loss across remaining keypoints, indicating that corpus is one of the most influential parts.



(c) Occluding the left side of the body improves the performance of right shoulder, ear and elbow.

Figure 5.8: Change in AP for top 5 keypoints with the largest difference, when chosen part is occluded.

5.4. OCCLUSION AUGMENTATION AGAINST ATTACKS

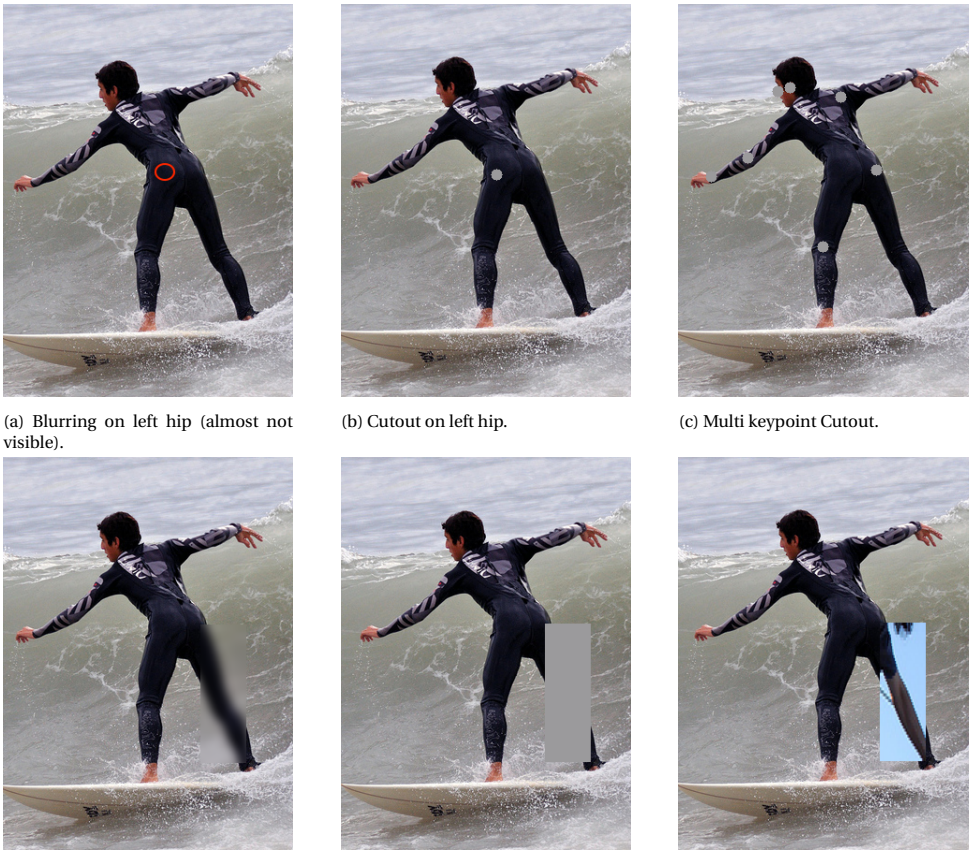
We evaluate two main human pose estimation datasets: COCO [45] has 200k images with 250k person instances, labelled with 17 keypoints and MPII [46] has 40k persons, each labelled with 16 joints. The train, validation and test sets include 22k, 3k and 15k person instances respectively. For the evaluation of MPII dataset, the validation set is used since the labels of the test set are not available.

For training HRNet [19] models on COCO [45] and MPII [46] we follow the original pipeline of HRNet. For COCO dataset, human detection boxes are extended to fit 4:3 aspect ratio, and cropped from the image and resized to $256x192$. The pose estimator is trained with the keypoint location of the joints. The data augmentations that are used in HRNet training include random rotation $\in [-45^o, 45^o]$, random scale $\in [0.65, 1.35]$, random flipping and half-body augmentations. The Adam optimizer [57] is used to train the

network with the learning rate schedule following [40], starting with $1e-3$ and reduced to $1e-4$ and $1e-5$ at 170th and 200th epochs respectively and the training is completed at the 210th epoch. For MPII dataset, the training procedure of HRNet is as followed: 256x256 input size is used and half-body augmentations are discarded. For the evaluation of the models, Object Keypoint Similarity (OKS) for COCO and Percentage of Correct Keypoints (PCK) for MPII are used.

During testing, HRNet firstly employs an object detection algorithm to obtain boxes with a single person. Afterwards the pose estimator produces the keypoint location of the joints.

5.4.1. OCCLUSION AUGMENTATION



(a) Blurring on left hip (almost not visible).

(b) Cutout on left hip.

(c) Multi keypoint Cutout.

(d) Part Blurring.

(e) Part Cutout.

(f) PartMix on the right leg with a pasted random arm.

Figure 5.9: Targeted keypoint augmentations: a, b, c and targeted part augmentations: d, e, f.

We investigate the following three methods: (i) Targeted Blurring, (ii) Targeted Cutout, (iii) Targeted PartMix. The augmentation techniques are called as *targeted*, because we

apply them on target locations of keypoints or parts instead of random location in the image. It is important to state that the proposed augmentation techniques are introduced after the bounding box person detection, and it thus does not affect the object detection method.

Targeted Blurring. We use Gaussian blur for two types of targeted blurring: (i) keypoint blurring with a kernel size of 9 pixels (Figure 5.9a) and (ii) part blurring with a kernel size of 31 pixels shown in Figure 5.9d.

Targeted Cutout. The size of the keypoint cutout (Figure 5.9b-5.9c) and part cutout (Figure 5.9e) are similar to the blurring equivalents. Instead of blurring, the area is colored with mean value of the image.

Targeted PartMix. The method is designed to mitigate the occlusions caused by another person (Figure 5.9f). In this approach, a different part from a random image is pasted in the place of a body part area. In this process, the keypoint labels of newly pasted part are not introduced to heatmap labels. This augmentation is only performed on body parts. Similar to the part level blurring and cutout augmentation methods, the occluded keypoints under the pasted area are still predicted.

5.4.2. ANALYSES OF OCCLUSION AUGMENTATION

All the following augmentation methods, except baselines, already include flipping, rotation, scaling and half-body augmentations. Each network obtains the boxes from Cascade RCNN [58] detector which has ResNet50 backbone. The results of each method can be seen in Table 5.1.

Baselines. Table 5.1 indicates 3 baseline variants. Firstly, HRNet without any augmentations obtains only 65.3% mAP score. Secondly, adding flipping, rotation and scaling augmentations improve non-augmented baseline by 8.6%. Last variant is half body augmentation which adds only 0.4% improvements on rotation and scaling augmentations.

Single keypoint augmentations. We check the performance of 3 different augmentations: blurring, cutout and a combination of two of them which are applied on a single keypoint with the varying probability of 0.2 and 0.5 (Figure 5.9a-5.9b). We observe the highest improvement for blurring and cutout by 0.2% when the probability is chosen as 0.5 (Table 5.1). Other single keypoint variants do not improve the performance.

Multi-keypoint augmentations. We applied random multi-keypoint variant blurring and cutout with a maximum of 5 keypoints with a probability of 0.2 (Figure 5.9c). The augmentation decreases the model performance by 0.4%.

Part augmentations. 4 different part augmentation methods are used: part blurring, part cutout, a combination of both them and PartMix (Figure 5.9d, 5.9e and 5.9f respectively). To demonstrate the effect of each augmentation, we apply them with a probability of 0.2 and 0.5. In addition, the effect of removing the labels of the occluded keypoint is also investigated as *removal* column in Table 5.1.

In the bottom part of Table 5.1, cutout and PartMix show 0.2% and 0.1% improvements respectively. In all the variants of blurring, small degradation or no improvement is observed. The combination of part level variants of cutout and blurring indicate some decreases of the performance for the removal configuration with probability of 0.2 and 0.5 and do not improve in non-removal configuration.

				Evaluation results					
Augmentation	*level*	*removal*	*p*	*AP*	AP^{50}	AP^{75}	AP^{M}	AP^{L}	*AR*
Baseline (no augments)	-	-	-	65.3	86.4	72.6	62.6	70.7	70.2
Baseline (flip, rot, scale)	-	-	-	73.9	90.0	80.9	70.4	80.3	78.3
Baseline (flip, rot, scale, half-body)	-	-	-	74.3	90.6	81.7	70.7	80.7	78.8
Blurring	k	✗	0.2	74.3	90.4	81.6	70.8	80.6	78.7
	k	✗	0.5	74.5	90.4	81.8	70.8	80.8	78.7
Cutout	k	✗	0.2	74.3	90.4	81.7	71.0	80.3	78.7
	k	✗	0.5	74.5	90.5	81.7	70.9	80.7	78.8
Cutout + Blurring	k	✗	0.2	74.0	90.4	81.1	70.4	80.3	78.4
	k	✗	0.5	74.3	90.5	81.1	70.8	80.6	78.6
Blurring	p	✔	0.2	74.3	90.5	81.7	70.6	80.8	78.6
	p	✔	0.5	74.0	90.5	81.1	70.5	80.4	78.4
	p	✗	0.5	74.1	90.3	81.1	70.6	80.2	78.5
Cutout	p	✔	0.2	74.2	90.5	81.2	70.8	80.4	78.6
	p	✔	0.5	74.2	90.3	81.1	70.6	80.4	78.6
	p	✗	0.5	74.5	90.5	81.6	70.9	80.7	78.8
Cutout + Blurring	p	✔	0.2	73.4	90.3	80.8	69.9	79.5	77.8
	p	✔	0.5	73.9	90.4	81.0	70.5	80.0	78.3
	p	✗	0.5	74.3	90.4	81.2	70.6	80.5	78.6
Multikeypoint (max. 5)	-	-	0.2	73.9	90.1	80.9	70.5	80.2	78.3
PartMix	-	✔	0.5	74.3	90.5	81.1	70.7	80.6	78.7
	-	✗	0.5	74.4	90.7	81.5	71.1	80.5	78.8

Table 5.1: Comparison of augmentation variants on COCO validation set for HRNet using CascadeRCNN bounding boxes. Upper-part indicates single-keypoint augmentation and bottom-part shows multiple-keypoint augmentation. k and p in the level column represent keypoint and part augmentations respectively. Removal column indicates if the occluded keypoints are removed from prediction. Column *p* is the probability of augmentation. Keypoint cutout and blurring, and part cutout and PartMix improve the performance. Other variants obtain results either on a par with baseline or worse than baseline.

To conclude to findings from the Table 5.1, flipping, rotation and scaling augmentations add a huge performance gain to the HRNet. However, including half-body, the occlusion based augmentation methods do not improve the performance of the pose estimator significantly.

The effect of the object detection algorithms. HRNet [19] is a top-down approach which utilizes an object detection algorithm to obtain human instances. Therefore, the performance of the pose estimation considerably depends on the detection performance, namely detected human instances.

By the evidence of the Table 5.1, we choose keypoint blurring, part cutout and PartMix methods for further analysis as they are the most promising augmentations.

We evaluate the pose estimation performances of vanilla HRNet and also of HRNet with the chosen augmentation methods with two 2-stage detectors, Faster RCNN [59] with XCeption 101 backbone and Cascade RCNN [58]; 2 single-stage detectors, RetinaNet [60] and EfficientDet D7 [61]; and by using ground truth boxes of human instances (Figure 5.10).

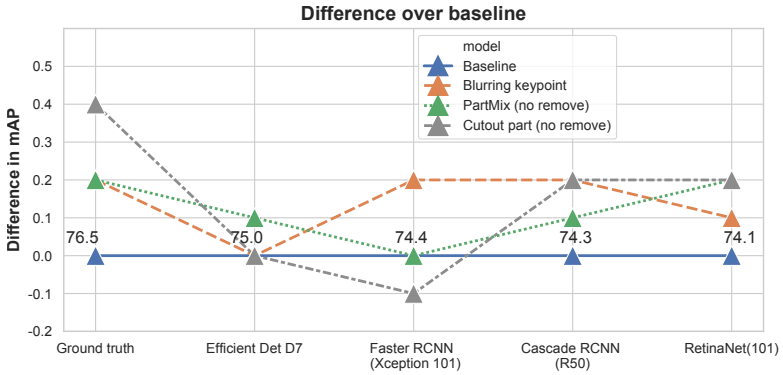


Figure 5.10: Performance of chosen augmentations for HRNet-32 on various detection backbones and ground truth boxes. The ground truth bounding box performs best. Yet, none of the data augmentation methods help to improve performance over 0.2% for any object detector.

				Evaluation results							
Augmentation	*level*	*remove*	*p*	*Head*	*Shoulder*	*Elbow*	*Wrist*	*Hip*	*Knee*	*Ankle*	***Total***
Baseline	-	-	-	97.1	95.9	90.4	86.4	89.1	87.2	83.3	90.3
Blurring	k	✗	0.5	97.3	95.9	90.5	86.2	89.2	86.4	83.1	90.3
Cutout	p	✗	0.5	97.2	96.3	90.7	86.7	89.4	86.7	83.3	**90.5**
PartMix	-	✗	0.5	97.4	96.2	91.0	86.8	89.2	86.7	83.0	**90.5**

Table 5.2: Results on MPII dataset. Keypoint blurring obtains on a par with the HRNet baseline, yet part cutout and PartMix increase the performance.

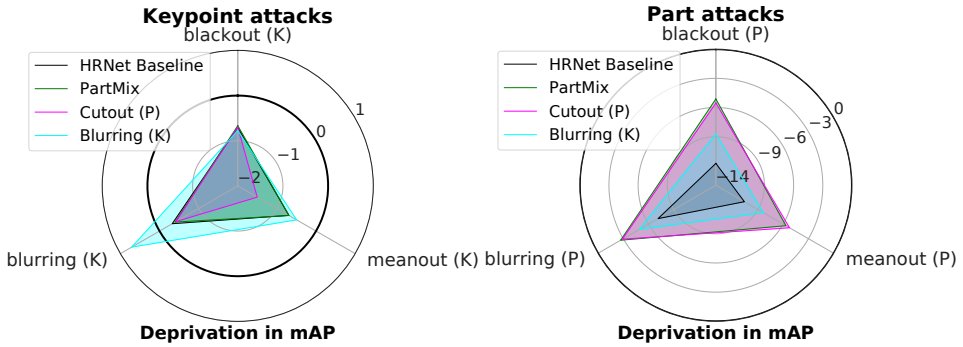
All the augmentations indicate improvements using ground truth bounding boxes by 0.2% for keypoint blurring and PartMix, and 0.4% for part cutout. All the chosen augmentation methods obtain better result with Cascade RCNN and RetinaNet 0.1 – 0.2% depending on the augmentation. With EfficientDet D7 detector, keypoint blurring and part cutout result in similar to baseline except 0.1% improvement by PartMix. For Faster-RCNN, keypoint blurring shows 0.2% increase, yet part cutout degrades the performance by 0.1%.

The performances of baseline and the augmentations vary depending on the object detector. The augmentation methods improves the results slightly, yet the gain is insignificant.

Performance on MPII. We also test the data augmentation methods on MPII dataset (Table 5.2). If we check the total contribution of the proposed augmentations, keypoint blurring result in on a par with baseline, yet part cutout and PartMix increase the performance by 0.2% for the metric PCK@0.5. The largest improvement per keypoint is observed for elbows by 0.6% and wrists by 0.3%, with the degradation on knees and ankles by 0.4% and 0.2% respectively.

Similar to analyses on the COCO dataset, the proposed augmentations can only improve the performance slightly.

How occlusion robust is data augmentation? Figure 5.11 shows the robustness of the baseline and the proposed augmentations to the occlusion attacks. The analysis is



(a) (b)

Figure 5.11: Robustness comparison of proposed methods against (a) keypoint and (b) part occlusion attacks. Part augmentations improve the baseline but does not solve occlusion.

done on COCO dataset and the results are shown as mAP score of all keypoints. We can clearly see that training with the keypoint blurring augmentation makes the network more robust against blurring attack, but there is no significant improvement for the other keypoint attacks. In case of part attacks, we observe an improvement across all augmentation methods over the baseline. For the part augmentations, there is a significant improvement against all part level attacks in comparison to baseline. Specifically, PartMix has almost no advantages against keypoint attacks, however, it improves part level methods about more than 5% in comparison to baseline. Part cutout obtains similar performance with PartMix against part attacks. Proposed augmentations reduce the performance deprivations when we apply occlusion attacks, yet data augmentation still does not solve the occlusion problem.

5.4.3. AUGMENTATION ON BOTTOM-UP METHOD: HIGHER HRNET

	Evaluation results				
Augmentation	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}
Higher HRNet	67.1	86.2	73.0	61.5	76.1
Blurring (K)	66.5	86.3	72.1	60.6	75.7
Cutout part (no remove)	66.6	86.4	72.9	60.7	75.6
PartMix (no remove)	67.0	86.4	73.0	61.3	75.8

Table 5.3: Results for bottom-up method, Higher HRNet [25]. The keypoint blurring, part cutout and Partmix degrade the performance of bottom-up methods. The augmentations do not help Higher HRNet.

We also apply occlusion augmentations on Higher HRNet [25], a bottom-up method. Higher HRNet is built on HRNet-32 and inputs 512x512 sized images. The training procedure follows Higher HRNet implementation from the paper. Unlike top-down methods, Higher HRNet operates on full-image and try to obtain the keypoints of each instance from the full-image. When applying the augmentations on Higher HRNet, we target all

the human instances in the image.

Results in Table 5.3 show the augmentation methods to improve AP50 score slightly. For AP, all augmentations degrade performance by 0.6% for keypoint level blurring, by 0.5% for part level cutout and by 0.1% for PartMix. Hence, using part and keypoint augmentations do not improve the performance of a bottom-up method.

5.5. DISCUSSION AND CONCLUSION

In this study, we investigate the sensitivity of human pose estimators to occlusion. Firstly, we introduce targeted keypoint and body part occlusion attacks to show how much occlusion affects the performance. Secondly, keypoint and part based data augmentation techniques against occlusion are investigated. The structured analyses indicate that deep pose estimators are not robust to occlusion. With all the bells and whistles, the current and proposed data augmentation methods do **not** bring significant improvements on the performance of the top-down pose estimators and even reduce the performance for the bottom-up approaches. Our paper is important because it helps data scientists looking for improvements against occlusions to not work on data augmentation. Battling occlusions is still an open problem for human pose estimation.

Part based attacks and augmentation are applied as a rectangle shape. This fact can introduce unusual artefacts because natural occlusions can have arbitrary shapes. Each keypoint augmentation is applied as a circle that covers the related keypoint, yet in reality, keypoint occlusions can occur with numerous shapes and ways e.g. self occlusion, occlusion by other object. Moreover, for bottom-up approaches, the input image into the network may have more perturbations since the full image can contain multiple instances. This fact can harm the learning process.

5.6. APPENDIX

5.6.1. ADDITIONAL RESULTS

HRNet results. For this experiment, we increase the input resolution of images from 256x192 to 384x256. The training process follows the aforementioned scheme for COCO dataset.

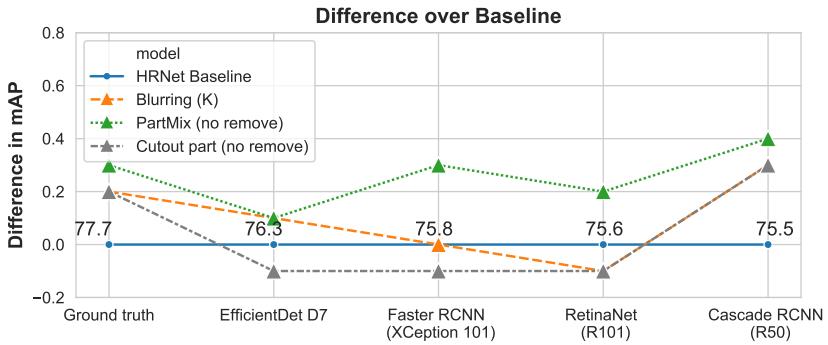


Figure 5.12: Higher resolution input for HRNet 32: the resolution is changed from 256x192 to 384x256. The best performance across detection backbones is observed for PartMix.

According to the analysis of the performance across a variety of detection backbones shown in Figure 5.12, we notice that PartMix is consistently improving performance - with the greatest boost of 0.4% for Cascade R-CNN and 0.3% for Faster RCNN. For both keypoint blurring and part cutout, we observe no significant improvement or even the performance decreases - for part cutout using EfficientDet, Faster RCNN and RetinaNet and for Blurring using RetinaNet. All the presented augmentations show largest gain for Cascade RCNN. Occlusion augmentations do not help to solve occlusion problems when higher resolution input is used.

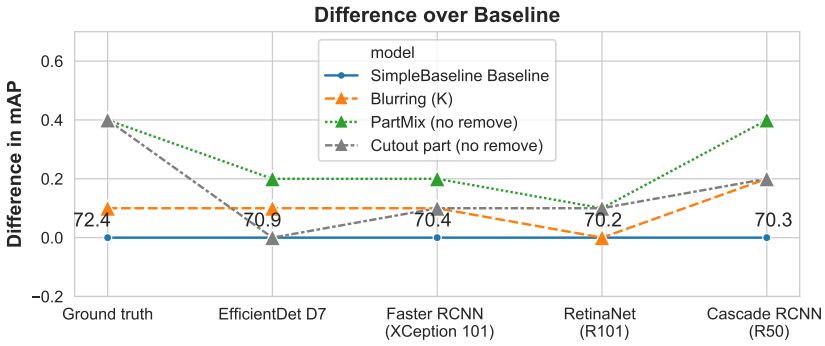


Figure 5.13: Performance of chosen augmentations for SimpleBaseline on various detection backbones and ground truth boxes. Using the ground truth bounding boxes outperforms all the SimpleBaseline methods with a detection backbones.

SimpleBaseline results. The usability of occlusion augmentations are not only limited to HRNet, yet they can be used with other top-down methods like SimpleBaseline [40]. In this experiment, we apply the occlusion augmentations on SimpleBaseline method with different object detection backbones. The training procedure of the network follows the original implementation.

By checking the performance across the various detection backbones we observe either small or no improvement at all (Figure 5.13). PartMix show the most significant improvement across detection backbones, with 0.4% boost in the performance for the ground truth boxes and the boxes produced by Cascade RCNN, 0.2% for EfficientDet and Faster RCNN and 0.1 % for RetinaNet. Cutout and Blurring improve at most 0.2% across all the detection backbones, apart from 0.4% for Cutout using ground truth bounding boxes. According to the results, proposed augmentation techniques do not solve occlusion problems of SimpleBaseline method.

5.6.2. VISUALIZATION OF RESULTS

Figure 5.14 presents a qualitative comparison between ground truth, HRNet-32 Baseline and keypoint blurring augmentation. In the first and second rows, keypoint blurring outperforms the baseline by obtaining the position of the left wrist and knee keypoints respectively. In the third row, both baseline and keypoint blurring produce wrong keypoint predictions. Fourth row presents failure case when baseline produces near-optimal annotations, while the method with keypoint blurring predicts left ankle in place of the right one.

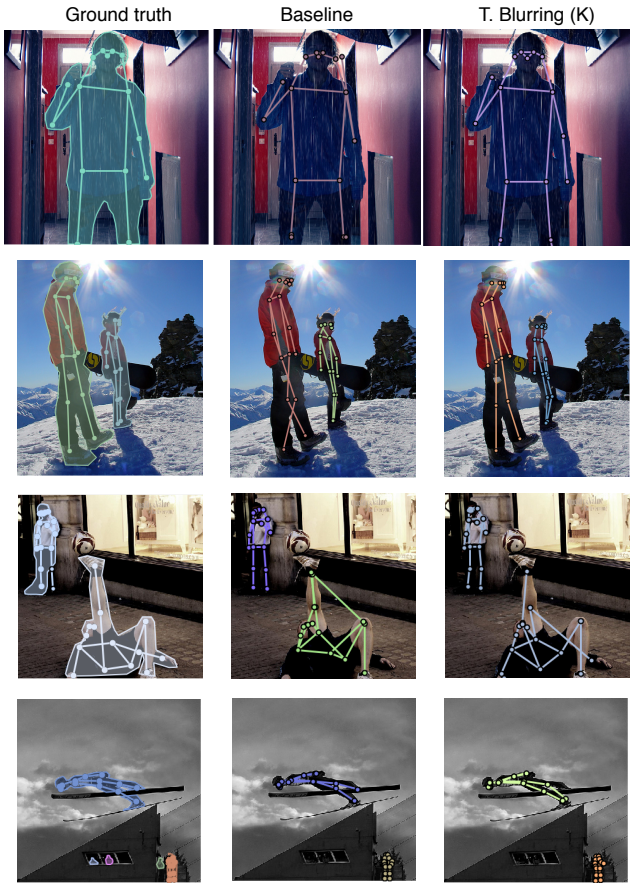


Figure 5.14: Qualitative comparison between ground truth (left), baseline (middle) and keypoint Blurring (K) (right). 1st and 2nd rows respectively - misplacement of left wrist keypoint and mismatch between knee keypoints in the baseline and keypoint blurring fixes the mistakes. 3rd row - both baseline and proposed method produce wrong keypoints. 4th row - baseline produces near-optimal keypoints whilst keypoint blurring makes mistake on left ankle keypoint. Data augmentation does not solve occlusion problem.

REFERENCES

[1] R. Pytel, O. S. Kayhan, and J. C. van Gemert, *Tilting at windmills: Data augmentation for deep pose estimation does not help with occlusions,* in *2020 25th International Conference on Pattern Recognition (ICPR)* (2021) pp. 10568–10575.

[2] Z. Li, X. Chen, W. Zhou, Y. Zhang, and J. Yu, *Pose2body: Pose-guided human parts segmentation,* in *2019 IEEE International Conference on Multimedia and Expo (ICME)* (IEEE, 2019) pp. 640–645.

[3] K. Lin, L. Wang, K. Luo, Y. Chen, Z. Liu, and M.-T. Sun, *Cross-domain complementary learning using pose for multi-person part segmentation,* IEEE Transactions on Circuits and Systems for Video Technology (2020).

[4] F. Xia, P. Wang, X. Chen, and A. L. Yuille, *Joint multi-person pose estimation and semantic part segmentation,* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

[5] D. Luvizon, D. Picard, and H. Tabia, *Multi-task deep learning for real-time 3d human pose estimation and action recognition,* IEEE Transactions on Pattern Analysis and Machine Intelligence , 1–1 (2020).

[6] X. Nie, C. Xiong, and S.-C. Zhu, *Joint action recognition and pose estimation from video,* 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 1293 (2015).

[7] K. Soomro, H. Idrees, and M. Shah, *Online localization and prediction of actions and interactions,* IEEE Transactions on Pattern Analysis and Machine Intelligence **41**, 459 (2019).

[8] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran, *Detect-and-Track: Efficient Pose Estimation in Videos,* in *CVPR* (2018).

[9] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, *Pose Flow: Efficient online pose tracking,* in *BMVC* (2018).

[10] A. Sokolova and A. Konushin, *Pose-based deep gait recognition,* CoRR **abs/1710.06512** (2017), arXiv:1710.06512 .

[11] H. L. Tavares, J. B. C. Neto, J. P. Papa, D. Colombo, and A. N. Marana, *Tracking and re-identification of people using soft-biometrics,* in *2019 XV Workshop de Visão Computacional (WVC)* (2019) pp. 78–83.

[12] Z. Fang and A. M. López, *Intention recognition of pedestrians and cyclists by 2d pose estimation,* ArXiv **abs/1910.03858** (2019).

[13] S. Perla, S. Das, P. Mukherjee, and U. Bhattacharya, *Cluenet : A deep framework for occluded pedestrian pose estimation,* (2019).

[14] S. Wang, F. Flohr, H. Xiong, T. Wen, B. feng Wang, M. Yang, and D. Yang, *Leverage of limb detection in pose estimation for vulnerable road users,* 2019 IEEE Intelligent Transportation Systems Conference (ITSC) , 528 (2019).

[15] P. A. Dias, D. Malafronte, H. Medeiros, and F. Odone, *Gaze estimation for assisted living environments,* in *The IEEE Winter Conference on Applications of Computer Vision* (2020) pp. 290–299.

[16] Š. Obdržálek, G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison, and M. Pavel, *Accuracy and robustness of kinect pose estimation in the context of coaching of elderly population,* in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (IEEE, 2012) pp. 1188–1193.

[17] L. Ladický, P. H. S. Torr, and A. Zisserman, *Human pose estimation using a joint pixel-wise and part-wise formulation,* in *2013 IEEE Conference on Computer Vision and Pattern Recognition* (2013) pp. 3578–3585.

[18] J. Varadarajan, R. Subramanian, S. R. Bulò, N. Ahuja, O. Lanz, and E. Ricci, *Joint estimation of human pose and conversational groups from social scenes,* International Journal of Computer Vision **126**, 410 (2018).

[19] K. Sun, B. Xiao, D. Liu, and J. Wang, *Deep High-Resolution Representation Learning for Human Pose Estimation,* (2019), arXiv:1902.09212 .

[20] T. Golda, T. Kalb, A. Schumann, and J. Beyerer, *Human pose estimation for real-world crowded scenarios,* CoRR **abs/1907.06922** (2019), arXiv:1907.06922 .

[21] Y. Huang, B. Sun, H. Kan, J. Zhuang, and Z. Qin, *Followmeup sports: New benchmark for 2d human keypoint recognition,* ArXiv **abs/1911.08344** (2019).

[22] U. Rafi, J. Gall, and B. Leibe, *A semantic occlusion model for human pose estimation from a single depth image,* in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2015) pp. 67–74.

[23] S. Shah, N. Jain, A. Sharma, and A. Jain, *On the robustness of human pose estimation,* (2019).

[24] C. Shorten and T. M. Khoshgoftaar, *A survey on image data augmentation for deep learning,* Journal of Big Data **6**, 60 (2019).

[25] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, *Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation,* in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[26] L. Taylor and G. Nitschke, *Improving deep learning using generic data augmentation,* arXiv preprint arXiv:1708.06020 (2017).

[27] T. Devries and G. W. Taylor, *Improved regularization of convolutional neural networks with cutout,* CoRR **abs/1708.04552** (2017), arXiv:1708.04552 .

[28] H. Guo, Y. Mao, and R. Zhang, *Mixup as locally linear out-of-manifold regularization,* CoRR **abs/1809.02499** (2018), arXiv:1809.02499 .

[29] C. Summers and M. J. Dinneen, *Improved mixed-example data augmentation,* in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2019) pp. 1262–1270.

[30] Y. Tokozume, Y. Ushiku, and T. Harada, *Between-class learning for image classification,* in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018) pp. 5486–5494.

[31] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, *Manifold mixup: Better representations by interpolating hidden states,* in *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 97, edited by K. Chaudhuri and R. Salakhutdinov (PMLR, Long Beach, California, USA, 2019) pp. 6438–6447.

[32] H. Xu, Y. Li, W. Chen, D. Lischinski, D. Cohen-Or, and B. Chen, *A holistic approach for data-driven object cutout,* CoRR **abs/1608.05180** (2016), arXiv:1608.05180 .

[33] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, *mixup: Beyond empirical risk minimization,* CoRR **abs/1710.09412** (2017), arXiv:1710.09412 .

[34] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, *Random erasing data augmentation,* CoRR **abs/1708.04896** (2017), arXiv:1708.04896 .

[35] J. Choe, S. Lee, and H. Shim, *Attention-based dropout layer for weakly supervised single object localization and semantic segmentation,* IEEE Transactions on Pattern Analysis and Machine Intelligence , 1 (2020).

[36] N. Dvornik, J. Mairal, and C. Schmid, *Modeling visual context is key to augmenting object detection datasets,* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018) pp. 364–380.

[37] K. K. Singh and Y. J. Lee, *Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization,* in *2017 IEEE International Conference on Computer Vision (ICCV)* (2017) pp. 3544–3553.

[38] G. French, T. Aila, S. Laine, M. Mackiewicz, and G. D. Finlayson, *Consistency regularization and cutmix for semi-supervised semantic segmentation,* CoRR **abs/1906.01916** (2019), arXiv:1906.01916 .

[39] L. Ke, M. Chang, H. Qi, and S. Lyu, *Multi-scale structure-aware network for human pose estimation,* CoRR **abs/1803.09894** (2018), arXiv:1803.09894 .

[40] B. Xiao, H. Wu, and Y. Wei, *Simple baselines for human pose estimation and tracking,* CoRR **abs/1804.06208** (2018), arXiv:1804.06208 .

[41] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, *Openpose: Realtime multi-person 2d pose estimation using part affinity fields,* CoRR **abs/1812.08008** (2018), arXiv:1812.08008 .

[42] S. Kreiss, L. Bertoni, and A. Alahi, *Pifpaf: Composite fields for human pose estimation,* CoRR **abs/1903.06593** (2019), arXiv:1903.06593 .

[43] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, *Cascaded pyramid network for multi-person pose estimation,* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).

[44] A. Newell, K. Yang, and J. Deng, *Stacked hourglass networks for human pose estimation,* in *ECCV* (2016).

[45] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft COCO: common objects in context,* CoRR **abs/1405.0312** (2014), arXiv:1405.0312 .

[46] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, *2d human pose estimation: New benchmark and state of the art analysis,* in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014).

[47] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, *Human pose estimation with iterative error feedback,* in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016) pp. 4733–4742.

[48] A. Toshev and C. Szegedy, *Deeppose: Human pose estimation via deep neural networks,* in *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014) pp. 1653–1660.

[49] X. Chu, W. Yang, W. Ouyang, C. Ma, A. Yuille, and X. Wang, *Multi-context attention for human pose estimation,* (2017) pp. 5669–5678.

[50] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, *Convolutional pose machines,* in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016) pp. 4724–4732.

[51] W. Li, Z. Wang, B. Yin, Q. Peng, Y. Du, T. Xiao, G. Yu, H. Lu, Y. Wei, and J. Sun, *Rethinking on multi-stage networks for human pose estimation,* ArXiv **abs/1901.00148** (2019).

[52] F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu, *Distribution-aware coordinate representation for human pose estimation,* ArXiv **abs/1910.06278** (2019).

[53] J. Huang, Z. Zhu, F. Guo, and G. Huang, *The devil is in the details: Delving into unbiased data processing for human pose estimation,* in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[54] N. D. Reddy, M. Vo, and S. G. Narasimhan, *Occlusion-net: 2d/3d occluded keypoint localization using graph networks,* in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019) pp. 7318–7327.

[55] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, *Cutmix: Regularization strategy to train strong classifiers with localizable features,* CoRR **abs/1905.04899** (2019), arXiv:1905.04899 .

[56] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, *Learning feature pyramids for human pose estimation,* CoRR **abs/1708.01101** (2017), arXiv:1708.01101 .

[57] D. Kingma and J. Ba, *Adam: A method for stochastic optimization,* International Conference on Learning Representations (2014).

[58] Z. Cai and N. Vasconcelos, *Cascade r-cnn: Delving into high quality object detection,* 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition , 6154 (2018).

[59] S. Ren, K. He, R. B. Girshick, and J. Sun, *Faster R-CNN: towards real-time object detection with region proposal networks,* CoRR **abs/1506.01497** (2015), arXiv:1506.01497 .

[60] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, *Focal loss for dense object detection,* 2017 IEEE International Conference on Computer Vision (ICCV) , 2999 (2017).

[61] M. Tan, R. Pang, and Q. V. Le, *Efficientdet: Scalable and efficient object detection,* ArXiv **abs/1911.09070** (2019).

6

t-EVA: Time-Efficient t-SNE Video Annotation

This chapter has been published as:
S. Poorgholi, O. S. Kayhan, and J. C. van Gemert, t-eva: Time-efficient t-sne video annotation, in International Conference on Pattern Recognition (Springer, 2021) pp. 153–169. [1]

ABSTRACT

Video understanding has received more attention in the past few years due to the availability of several large-scale video datasets. However, annotating large-scale video datasets are cost-intensive. In this work, we propose a time-efficient video annotation method using spatio-temporal feature similarity and t-SNE dimensionality reduction to speed up the annotation process massively. Placing the same actions from different videos near each other in the two-dimensional space based on feature similarity helps the annotator to group-label video clips. We evaluate our method on two subsets of the ActivityNet (v1.3) and a subset of the Sports-1M dataset. We show that t-EVA[1] can outperform other video annotation tools while maintaining test accuracy on video classification.

[1] https://github.com/spoorgholi74/t-EVA

6.1. INTRODUCTION

The availability of large-scale video datasets [2–4] has made video understanding in various tasks such as action recognition [5–7], object tracking [8–10] an attractive topic of research. Various supervised methods [6, 7, 11] have improved video classification and temporal localization accuracy on large-scale video datasets such as ActivityNet (v1.3) [2]; however, labeling videos on such a large-scale dataset, requires a great deal of human effort. Therefore, other methods aim to train the networks for tasks such as video action recognition in a semi-supervised [12, 13] manner without having the full labels. To decrease the dependency on the quality and amount of annotated data, [14, 15] investigate pre-training features with internet videos with noisy labels in a weakly supervised manner. However, these methods do not achieve higher accuracy on video classification tasks than supervised models on large-scale video datasets such as Kinetics [3]. Instead of using such techniques, we focus on reducing the annotation effort for adding more training data.

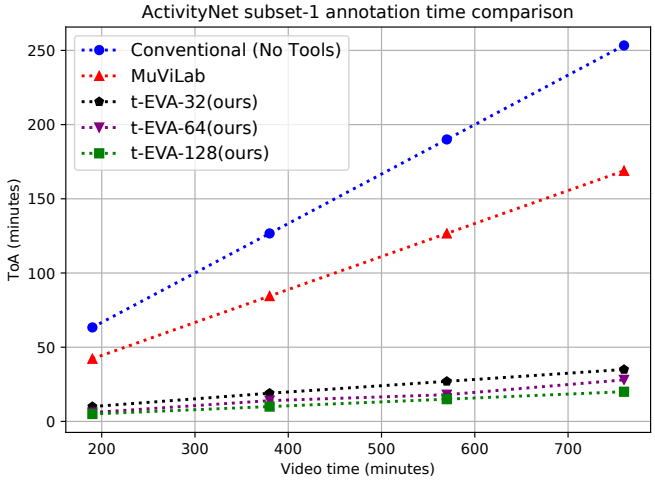


Figure 6.1: Comparison of annotation time using different tools versus video time for the ActivityNet [2] subset-1. Our annotation method (t-EVA) outperforms the conventional (no specific tools) annotation and MuViLab [16] in annotation time. With a window size of 128 time-steps (128-TS), our method can annotate 769 minutes of video in 21 minutes. The MuViLab and conventional annotation numbers are extrapolated.

Fully-supervised models require much annotated data that is unavailable as videos are unlabeled by nature, and annotating them is labor-intensive. Large scale datasets [2, 3, 17] use strategies like *Amazon Mechanical Turk* (AMT) to annotate the videos. [3] uses majority voting between multiple AMT workers to accept annotation of a single video. Using such methods is not efficient for video annotation on a large scale as it costs a lot in terms of time and money. MuViLab [16], an open-source software, enables the oracle to annotate multiple parts of a video simultaneously. However, these methods do not exploit the structure of the video data.

We introduce an annotation tool that helps the annotator group-label videos based

on their latent space feature similarity in a 2-dimensional space. Transferring the high-dimensional features obtained from 3D ConvNet to two dimensions using t-SNE gives the annotator an easy view to group label the videos both, temporal labels and classification labels. The annotation speed depends on the quality of the extracted features and how well they are placed together in the t-SNE plot. If the classes are well-separated in the t-SNE plot, group labeling becomes faster for the oracle.

We evaluate our method on two subsets of ActivityNet (v1.3 datasets)[2] and a subset of Sports-1M dataset [4] with 15 random classes. *Conventional annotation* refers to humans watching the videos and annotating the temporal boundaries of the human actions in videos without any specific tool. *MuViLab* is a more advanced open-source tool that extracts short clips from each video and plays them simultaneously in a grid-like figure beside each other. Oracle can annotate the video by selecting multiple short clips at the same time and assigning the specific class. We show that t-EVA outperforms conventional annotation techniques (with no specific tools) and MuViLab [16] in time of annotation (ToA) by a large margin on the ActivityNet dataset while still being able to keep the test accuracy on video classification task within a close range of using the original ground truth annotations (Figure 6.1).

6.2. Related Work

Video Understanding. In the past, the focus was on the use of specific hand-designed features such as HOG3D [18] SIFT-3D [19], optical flow [20] and iDT [21]. Among these methods, iDT and Optical flow is being used in combination with CNNs in different architectures such as two-stream networks [22]. Afterwards, some methods use 2D CNNs and extract features from video frames and combine them with different temporal integration functions [23, 24]. The introduction of 3D convolutional [6, 25] in CNNs which extend the 2D CNNs in temporal dimension show promising results in the task of action recognition in large-scale video datasets. 3D CNNs in different variations such as single stream and multiple-stream are among state of the art in the task of video understanding [26–32]. In this paper, we utilize single a stream 3D CNN architecture to obtain video features.

Dimensionality Reduction. Dimensionality reduction (DR) is an essential tool for high-dimensional data analysis. In linear DR methods such as PCA, the lower-dimension representation is a linear combination of the high-dimensional axes. Non-linear methods, on the other hand, are more useful to capture a more complex high-dimensional pattern [33]. In general, non-linear DR tries to maintain the local structure of the data in the transition from high-dimension to low-dimension and tends to ignore larger distances between the features [34]. t-Distributed Stochastic Neighbor Embedding (t-SNE) introduced by [35] is a non-linear DR technique which is used more for visualization. [36] shows that t-SNE is able to distinct well-separable clusters in low-dimensional space. Moreover, some works [34, 37, 38] propose for more effective use of t-SNE . [34] proposes a tool to support interactive exploration and visualization of high-dimensional data. An alternative to t-SNE is UMAP [37]; however, t-SNE is well studied, shows good results, and has the benefit of high-speed optimization [38]. t-EVA uses t-SNE to reduce the dimensionality of the feature representations.

Data Annotation is essential for supervised models. Different tools are proposed for

making an easy annotation tool for videos and images. However, they usually do not exploit the structure of the data, which is especially useful in videos [16, 39, 40]. Some methods [41–44] are designed to make the process of image annotation easier. [41] offers a real-time framework for annotating internet images, and [42] uses multi-instances learning to learn the classes and image attributes together; however, none of these methods use a deep representation of data. In more recent works, [43] utilizes *Deep Multiple Instance Learning* to automatically annotate images and [44] uses semi-supervised t-SNE and feature space visualization in lower dimension to provide an interactive annotation environment for images. [45] proposes a general framework for annotating images and videos. However, to the best of our knowledge, our method is the first video annotation platform that can *exploit the structure* of video using latent space feature similarity to increase the annotation speed.

6.3. T-EVA FOR EFFICIENT VIDEO ANNOTATION

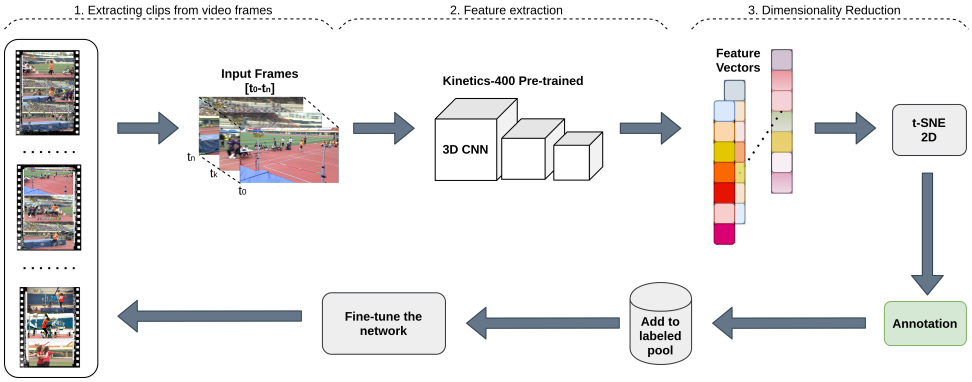


Figure 6.2: t-EVA pipeline: 1) Video clips are extracted from n consecutive frames [t_0-t_n] (time-steps). 2) Spatio-temporal features are extracted from the last layer of a 3D ConvNet before the classifier layer. 3) High dimensional features are projected to two dimensions using t-SNE and are plotted on a scatter plot. 4) Oracle annotates the clips represented in the scatter plot using a lasso tool. 5) The newly annotated data is added to the labeled pool. 6) The network is fine-tuned for a certain number of epochs. This cycle is repeated until all the videos are labeled, or the annotation budget runs out.

We propose incremental labeling with t-SNE based on feature similarity (Figure 7.2). First, several videos are randomly selected from the unlabeled pool, and 3D ConvNet features are extracted. The feature embeddings are transferred to a two-dimensional space using t-SNE. As it can be seen in Figure 6.3, the oracle has two subplots for annotation: (i) A plot in which the oracle can use a lasso tool to group label videos and (ii) Other plot with the middle frame of each clip in which the oracle can move and zoom with the cursor on the plot and observe where to annotate. After annotating the first set of videos, the video clips are moved to the labeled pool, and the 3D network is fine-tuned for a certain number of epochs with the newly labeled videos. We continue this process until all the videos are labeled, or the annotation budget finishes.

We use 3D ConvNets to extract features from the videos and split each video v into k shorter clips $v_i = [clip_1, ..., clip_k]$ by sampling every n non-overlapping frames $clip_i =$

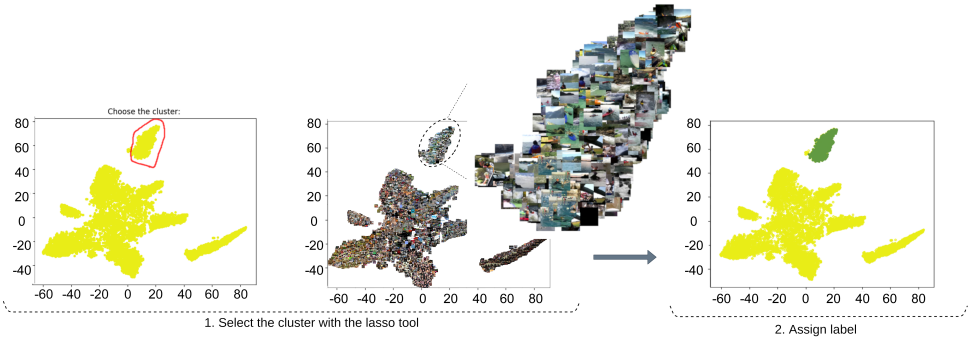


Figure 6.3: A minimal representation of the annotation tool. 1) The oracle can see the scatter plot (left) and the corresponding frames from the videos (middle) in separate figures. 2) Based on the figures' inspection, the oracle can detect different clusters of an action class (kayaking) and use the lasso tool to select the cluster. 3) In the end, the oracle assigns a label and based on the assigned class name, the selected points in the scatter plot change color.

$[frame_1, ..., frame_n]$. Sampling in multiple time-steps enables us to capture different lengths of actions in the dataset. Afterwards, each clip c_i is fed into the 3D ConvNet, for feature extraction. The features are extracted from the last convolution layer after applying global average pooling. In t-SNE, the pair-wise distances between feature vectors are used to map features to 2D.

In this paper, we use the Barnes-Hut optimized t-SNE version [46], which reduces the complexity of $O(NlogN)$ where N is the number of data-points.

6.3.1. How to Annotate?

An overview of the annotation procedure can be seen in Figure 6.3. First, the oracle sees the scatter plot with all points with the same color representing the unlabeled pool (Figure 6.3 left) and the corresponding middle frame of each clip in the video (Figure 6.3 middle). The oracle can move the cursor and zoom in the plot to inspect the frames with more details. Second, using the lasso tool, the oracle can draw a lasso around the scatter plots based on the visual similarity and inspection of the video frames. Third, oracle assigns the labels, and the network is fine-tuned for a certain number of epochs. The same process repeats until all the videos are annotated, or the annotation budget ends.

6.4. Experiments

In this section, we first explain the benchmark dataset and evaluation metrics. In addition, we empirically show how our t-EVA method can speed up annotation for the ActivityNet dataset while keeping the video classification accuracy in a close range to the usage of the ground truth labels. We also compare our results with MuViLab [16] annotation tool. Furthermore, we qualitatively show how t-EVA can help to annotate the Sports1-M [4].

6.4.1. DATASETS

ActivityNet (v1.3) is an untrimmed video dataset with a wide range of human activities [2]. It comprises of 203 classes with an average of 137 untrimmed videos per class in about 849 hours of video. We use two subsets of the ActivityNet dataset. The first subset comprises 10 random classes, namely *preparing salad, kayaking, fixing bicycle, mixing drinks, bathing dog, getting a haircut, snatch, installing carpet, hopscotch, zumba* consisting of 607 videos with 407 training videos and 200 testing videos. The second subset adds another 5 handpicked classes, which are *playing water polo, high jump, discus throw, rock climbing, using parallel bars*, and they are visually close to some of the 10 random classes to make the classification task harder. The second subset comprises 950 videos with 639 videos in training and 311 videos in the test set.

Sports-1M is a large-scale public video dataset with 1.1 million YouTube videos of 487 fine-grained sports classes [4]. We choose a subset of 15 random classes of the Sports-1M dataset, namely *boxing, kyūdō, rings (gymnastics), yoga, judo, skiing, dachshund racing, snooker, drag racing, olympic weightlifting, motocross, team handball, hockey, paintball, beach soccer* with 702 videos in total. The dataset provides video level annotation for the entire untrimmed video; however, the temporal boundaries of the actions in the video are not identified. Approximately 5% of the videos contain more than one action label.

6.4.2. EVALUATION METRICS

To evaluate our method on ActivityNet subsets, we report the *time of annotation* (ToA) as a metric to measure how fast the oracle can annotate a certain number of videos. The ToA score is an average of three times repeating each experiment by the oracle. ToA for conventional annotation and MuViLab on ActivityNet subset-1 is extrapolated since annotating 13 hours of video using these methods is not feasible. We also report video classification accuracy in the form of mean average precision (mAP) for the ActivityNet subsets to measure the quality of annotation when the network is fine-tuned with our annotations versus with the ground truth annotations. mAP is used instead of a confusion matrix since some videos of ActivityNet contain more than one action [2].

For the Sports-1M [4] dataset, we perform a qualitative analysis of the t-SNE projections. To motivate our design choices beyond qualitative results, we introduce a realistic annotation emulation metric to estimate the quality of t-SNE projections on a global and local level. To report how well the t-SNE projection can separate the classes at a global level, we use a measure of cluster homogeneity, and completeness. Homogeneity measures if the points in a cluster only belong to one class and completeness measures if all points from one class are grouped in the same cluster. In an ideal t-SNE projection, all the points in each cluster belong to one class (homogeneity=1.0), and all the points from a class are in the same cluster (completeness=1.0), which makes the annotation process much faster. For clustering, K-Means clustering with K being the number of classes is used. We use the K-Means clustering algorithm because it is fast and has less hyperparameters to choose.

Since ToA can be a subjective metric, to evaluate the generalization of t-EVA and to emulate the oracle's annotation speed better, we also use a measure of local homogeneity using K-nearest neighbors (KNN) with K=4 as in [44].

KNN can be used to estimate the local homogeneity between the features in lower di-

mensions. Higher KNN accuracy results in higher local homogeneity and better grouping; namely, the oracle can annotate the videos faster.

6.4.3. Implementation Details

Feature Extraction. We use the 3D ResNet-34 architecture [47], pre-trained on Kinetics-400, as a feature extractor for all the experiments owing to their good performance and usage of RGB frames only. As in [47], each frame is resized spatially to 112×112 pixels from the original resolution. Each video is transferred to clips by sampling every 32 consecutive frames. The feature extractor in every forward pass takes a clip in the form of a 5D tensor as an input. Each dimension of the input tensor represents the batch size, input color channels, number of frames, spatial height, and width, respectively. Namely, an input tensor for a clip sampled at 32 frames can be shown as (1, 3, 32, 112, 112). The features are extracted after the final 3D average pooling with an 8x4x4 kernel before the classifier layer. The dimensions of the feature vectors are k×512 with k being the total number of clips and later reduced to k×2 using t-SNE.

t-SNE. For dimensionality reduction, a Barnes-Hut implementation of t-SNE with two components are used from the scikit-learn library [48]. The perplexity is set to 30, and the early exaggeration parameter is 12, with a learning rate of 200. The cost function is optimized for 2500 iterations.

Training. After annotating each set of videos, the network is fine-tuned for a certain number of epochs. For training, the same 3D ResNet-34 [47] architecture is used. The sample duration is chosen as 32 frames for each clip, and the input batch size is 32. Stochastic gradient descends (SGD) is used as the optimizer with a learning rate of 0.1, weight decay of 1e-3, and momentum of 0.9.

6.4.4. Results on ActivityNet

ActivityNet Subset-1. First, we put all the 407 videos in the unlabeled pool. Then, we divide the videos randomly into four different sets of unlabeled videos. The clips are generated with 32 consecutive frames, and the features are extracted using the 3D Resnet-34. After annotating each set of unlabeled videos, the network is fine-tuned for 20 epochs with the labeled videos. To note that, previously labeled videos are also used in the later epochs. The process continues until the network reaches 100 epochs. Between epoch 60 and 100, the network is fine-tuned using all 407 videos. Meanwhile, we refine the labels of the videos.

The videos are annotated incrementally, each time one set is labeled. Table 6.1 shows that the annotation time drops after every iteration of annotation and fine-tuning. Before fine-tuning the network, the labeling of the first set takes 600 seconds. ToA reduces 150 seconds at epoch 60 when the network is fine-tuned with previously labeled videos. Because of the incremental labeling and fine-tuning, the network learns to extract better features from the videos, which can be better grouped in the t-SNE plot. It is also expected that the oracle spends more time annotating the first few unlabeled set as the network is not yet fine-tuned. The quality of annotation at the early stage significantly impacts the next iterations of extracted features.

Annotation Speed. To evaluate the annotation speed, we choose three methods: conventional, MuViLab [16], and t-EVA.

Table 6.1: Oracle's time of annotation (ToA) is shown on subset 1 of the ActivityNet (v1.3) dataset with 10 classes containing 407 videos (~13 hours). At every 20 iterations from 0 to 60, 102 new videos are annotated, and the network is fine-tuned for 20 epochs. From epoch 60 to 100, no new video is added. The previous video labels are refined by the oracle as the network can extract better features. The network is fine-tuned on the existing labeled videos until epoch 100. It can be seen with incremental annotation and fine-tuning the annotation time in the later epochs drops.

Epoch	0	20	40	60	80	100
ToA (seconds)	600	552	516	450	240	180

One way to increase the annotation speed of t-EVA is by putting more videos on the screen for the oracle to annotate. However, it does not make the labelling process easier. Since ActivityNet videos on average have 30 frames per second (FPS), every 32 time-steps that we sample represent almost 1 second ($\sim \frac{32}{30}$) of video. Putting all of the 407 videos (13 hours) overflows the screen with the frames and makes the annotation harder for the oracle. One way to prevent overflowing the figures with thousands of frames is to increase the time-steps for sampling frames from each clip to the point that the network can still preserve the clips' temporal coherency. This way, we can show all of the videos on the 2D plot with fewer points. Consequently, we design three different t-EVA in terms of the number of time steps as t-EVA-32, 64, and 128.

Table 6.2: Comparison of time gain when annotating with different methods on a subset-1 of ActivityNet containing 769 minutes of video. Our method (t-EVA) with 128 time-steps outperforms conventional, and MuViLab [16] methods with labeling 769 minutes of video in 21 minutes. Using more consecutive frames increases annotation speed.

	Conventional	**MuViLab**	**t-EVA-32**	**t-EVA-64**	**t-EVA-128**
Time Gain	3 x	4.5 x	18 x	24 x	36 x

First, we choose ActivityNet subset-1 with a total duration of 769 minutes. We annotated 30 minutes of videos using MuViLab and Conventional methods and extrapolated the result to match the total duration of ActivityNet subset-1. Additionally, the entire subset-1 is annotated using different variants of t-EVA, and we compare the annotation speed of all these methods (Table 6.2). The results show that labeling 769 minutes of video takes approximately 42 minutes with the t-EVA-32 method. t-EVA-32 outperforms both conventional and MuViLab methods on ActivityNet subset-1 in annotation speed by a large margin by respectively 4 to 6 times faster. With t-EVA-64 and 128, time gain can reach respectively 24 and 36 times more. Conventional annotation and MuViLab do not take advantage of the temporal dimension of videos for annotation. Nevertheless, our method exploits the spatio-temporal features and places similar actions near each other in the t-SNE plot for the oracle to annotate the actions.

We also evaluate the performance of the network on the test set of ActivityNet subset-1. In Figure 6.4, we compare the classification performance of the networks: (i) fine-tuned with original ground truth labels and (ii) fine-tuned by using newly annotated videos by 32, 64, and 128 time-steps. Annotating the videos with t-EVA method can

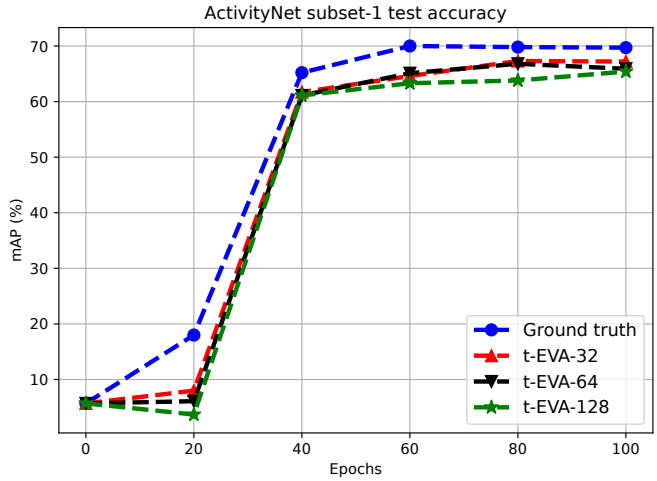


Figure 6.4: Comparison of video classification performance in the form of mAP (%) between fine-tuning the 3D ConvNet on ground truth label versus fine-tuning with our annotation acquired using different time-steps (TS). Fine-tuning the 3D ConvNet on the annotation generated by our method can achieve comparable video classification accuracy to the ground truth.

achieve a classification performance of 67.2% with 32-TS, 65.9% with 64-TS, and 65.4% with 128-TS, which is comparable to the training with ground truth labels (blue) by 69.7% mAP.

Table 6.3 shows the speed-accuracy trade off between t-EVA and ground-truth annotation. When the original ground truth labels are used for fine-tuning the network, we obtain 69.7% of mAP. 407 videos can be labeled in 42 minutes with t-EVA-32 by losing only 2.5% of performance in comparison to using ground truth labels. When the time-steps are increased as 64 and 128, the annotation speed decreases respectively to 31 and 21 minutes, yet the classification performance also reduces by 3.8% and 4.3%. Using 128 time-steps (t-EVA-128) reduces test accuracy while increasing the annotation speed. The decrease in accuracy compared to the 32-TS version is expected since the annotation is more prone to noise when the time-step is increased to 128 frames. With 128-TS for each clip, every point in the scatter plot represents 4 seconds of the video while it represents 1 second in the 32-TS version. Namely, labeling points wrongly in the 128 version (t-EVA-128) brings more significant consequences in the fine-tuning process. However, Table 6.3 indicates that using 128-TS (t-EVA-128) compared to the 32-TS (t-EVA-32) increases the annotation speed twice while the mAP score decreases less than 2%.

6.4.5. GENERALIZATION

To further demonstrate the generalization of our method, we conduct the same annotation experiment on a more challenging subset of ActivityNet (v1.3) with 15 classes and a subset of Sports-1M [4] with 15 random classes.

ActivityNet (v1.3) Subset-2. Subset 2 of ActivityNet (v1.3) contains 637 training videos

Table 6.3: Comparison of video classification performance (mAP) and ToA (time of annotation) on ActivityNet subset-1. This subset contains 407 videos in about 13 hours of video. Our method in 32 time-steps (t-EVA-32) and 128 time-steps (t-EVA-128) achieves comparable test accuracy to the ground truth accuracy and requires a much shorter time to annotate. There is a trade-off between annotation speed and performance.

Method	**GT**	**t-EVA-32**	**t-EVA-64**	**t-EVA-128**
mAP	69.7 %	67.2 %	65.9 %	65.4 %
ToA (minutes)	-	42	31	21

and 311 test videos. The first iteration of features is extracted from the 637 training videos and is annotated in 15 minutes by the oracle using t-EVA. After 20 epochs of fine-tuning, the new features are extracted, and the labels are fine-tuned again by the oracle. After this stage, the network is fine-tuned for 80 epochs. After fine-tuning for 100 epochs, our method reaches a test accuracy of 66.4%, while the training with ground-truth labels achieves an accuracy of 68.3% on the video classification task.

The 4-NN accuracy of the final features is 92.4%, which shows the quality of the extracted features is sufficient for the oracle to annotate. t-EVA can also perform well on the ActivityNet subset-2. The fact validates that our method can also generalize on a more challenging subset of ActivityNet.

Sports-1M. We further validate our method on a subset of Sports-1M [4] dataset with 15 random classes. We randomly sample 200 videos (~860 minutes) from the total 702 videos available in the 15 classes. The features are extracted from 200 videos, and ground truth labels of the two-dimensional features can be seen in Figure 6.5. Using 4-NN, we obtain an accuracy of 92.3%, which shows the features can be annotated based on similarity. Using our method, we were able to annotate 860 minutes of video in 28 minutes, giving us a time gain of 30.7. t-EVA indicates an extensive time gain on the Sports-1M dataset.

6.5. ABLATION STUDY

In this section, we conduct an ablation study to motivate our design choices in the following aspects: (i) dimensionality reduction method, (ii) t-SNE parameter selection, and (iii) 2D versus 3D backbone for feature extraction.

6.5.1. DIMENSIONALITY REDUCTION

We investigate using PCA as a linear dimensionality method and t-SNE as a non-linear dimensionality method for visualizing the high-dimensional features in two dimensions. We use the extracted feature from the ActivityNet subset-1 with 407 videos. Figure 6.6-b shows that qualitatively PCA is not able to group similar features and separate unalike features from the videos in the transition to a lower dimension, making the annotation more difficult. However, Figure 6.6-a indicates that t-SNE projection can maintain the local structure of each class while separating the features from different classes. To report the quality of projection in quantitative measures, we use KNN with K=4. The 4-NN classification accuracy in Figure 6.6 for the t-SNE projection is 80.6%, and for the PCA projection is 58.2%. Therefore, PCA, a linear dimensionality method, cannot reduce the

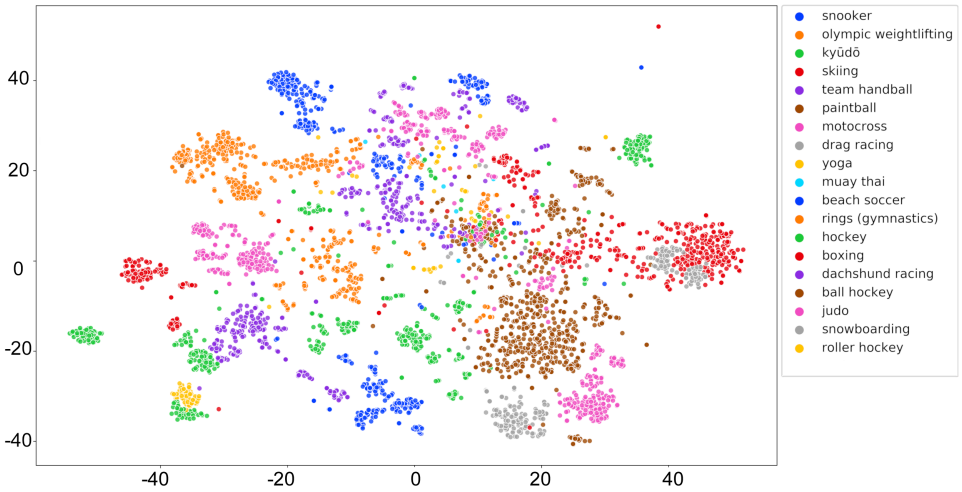


Figure 6.5: t-SNE projection of extracted features from 200 videos from the Sports-1M [4] dataset with ground truth labels as colors. 200 videos are from 15 random classes; however, some videos contain more than one activity class. The 4-NN accuracy, which emulates the quality of the projection through measuring local homogeneity, is 92.3%, indicating such a figure is annotate-able by the oracle.

feature dimension while placing similar classes near each other.

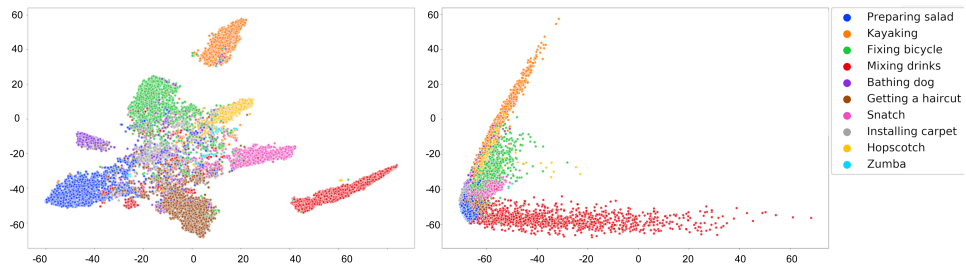


Figure 6.6: Visual comparison of the projection quality of high-dimensional features to two dimensions using t-SNE (a) and PCA (b). PCA is unable to maintain the structure of the high-dimensional data in two dimensions.

6.5.2. T-SNE PARAMETERS

We investigate using different perplexity parameters for the t-SNE projection. [35] recommend using perplexity parameter between [5-50], however larger and denser datasets requires relatively higher perplexity. With low perplexity, the local structure of data in each video dominates the action grouping from multiple video [49], but our goal is to

Table 6.4: Comparison of homogeneity and completeness scores as a measure to emulate the quality of t-SNE projection on a global-level. Higher homogeneity means all the points in a cluster belong to the same class. Higher completeness means all the points belonging to a class are in the same cluster. t-SNE perplexity parameter as 30 gives the highest homogeneity and completeness score.

	px-5	**px-15**	**px-30**	**px-50**	**px-100**	**px-120**
Homogeneity	44.7%	58.7%	62.5%	61.3%	61.7%	61.5%
Completeness	42.5%	56.1%	60%	58.5%	59%	58.8%

group multiple actions from different videos. To emulate the t-SNE projection quality for the annotation, we report homogeneity and completeness scores with different perplexities in Table 6.4. Perplexity 30 shows the highest homogeneity and completeness scores, in other words, t-SNE projection with perplexity 30 can separate the classes better than projecting with the other perplexity parameters. Therefore, using t-SNE with perplexity 30 makes the group labeling process easier for the oracle.

6.5.3. 2D-3D COMPARISON

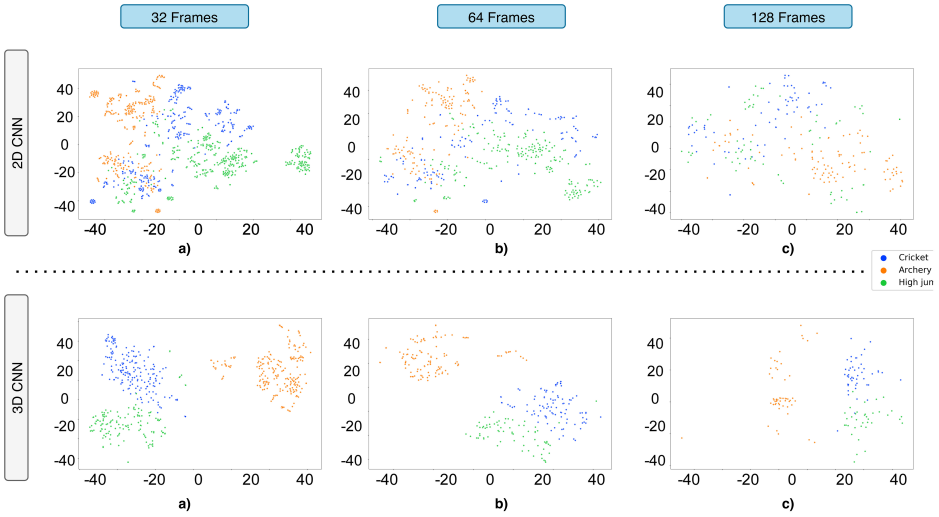


Figure 6.7: Comparison of t-SNE projection of extracted features from a 2D CNN versus a 3D ConvNet for videos from 3 action classes of ActivityNet dataset [2]. Increasing the time-steps for sampling clips from the videos causes the 2D CNN to lose the spatial information of the clips. However, the features from the 3D ConvNet can maintain the coherency between the clips.

We investigate replacing the 3D ConvNet with a 2D CNN to compare the quality of the feature embedding. For 3D ConvNet, 3D ResNet-34 pre-trained on Kinetics [3] and for the 2D CNN ResNet-50 pre-trained on Kinetics [3] are used. We chose Resnet-50 instead of Resnet-34 for the 2D CNN because the Kinetics pre-trained weights were only available for ResNet-50. To experiment, we sample every 32 consecutive frames (time-

steps) as a clip in the 3D ConvNet, and for the 2D CNN, we choose one frame for every 32 frames to represent that specific window. The experiment is done on the subset-1 of the Activity-Net dataset with 10 classes. It can be seen in Figure 6.7 that we start the experiments with 32 time-steps. With 32 time-steps, we can see the 2D CNN can capture the same action in different videos but can not place them together as well as the 3D ConvNet. Therefore, the colors representing the classes are better gathered nearby in the 3D ConvNet, making the annotation process faster than the 2D CNN projection. Moreover, by increasing the time-steps for frame sampling, the 2D CNN, even with deeper architecture, starts losing the temporal coherency between the data-points because 2D CNN only focuses on the spatial information between the frames. Focusing only on spatial information can still work in lower time-steps (32-TS) since the frames from the same action contain similar spatial information. However, using spatial information alone becomes problematic in higher time-steps as increasing the time-steps reduces the spatial similarity between the frames.

Table 6.5: Comparison of 4-NN accuracy of extracted features from a 2D CNN (ResNet-50) and a 3D ConvNet (3D ResNet-34) on subset-1 of ActivityNet [2]. Increasing time-steps cause the 2D CNN to lose the spatial similarity between the frames and fail to group them in the t-SNE plot, while the 3D ConvNet can still group similar actions even in higher time-steps.

	32-TS	**64-TS**	**128-TS**
2D CNN	93.1 %	89.3 %	74.6 %
3D CNN	100 %	97.6 %	95.2 %

To evaluate our findings quantitatively, we use K-NN accuracy as a quantitative emulation for the quality of features for annotation. Table 6.5 shows that increasing the number of frames in the clips degrades the 4-NN accuracy of 2D CNN dramatically from 93% to 75%. However, 3D CNN only loses around 5% from 32 time steps to 128. The local homogeneity decreases more drastically in 2D CNNs compared to 3D CNNs, which makes annotation more difficult for the oracle. In other words, the 2D CNN alone can not maintain the temporal structure of the data in higher time-steps. Thus, in the t-EVA method, 3D features are extracted to use for group labeling.

6.6. CONCLUSION

This paper introduced a smart annotation tool, t-EVA, for helping the oracle to group label videos based on their latent space feature similarity in two-dimensional space. Our experiments on subsets of large-scale datasets shows that t-EVA can be useful in annotating large-scale video datasets, especially if the annotation budget and time are limited. Our method can outperform the conventional annotation method, and MuViLab [16] time-wise in the order of magnitude with a minor drop in the video classification accuracy. Besides, t-EVA is a modular tool, and its components can be easily replaced by other methods. To illustrate, 3D ResNet can be changed to another feature extractor.

t-EVA method has a trade-off between annotation speed and network performance. Increasing time steps can reduce the annotation time; however, the network's accuracy may also decrease.

t-EVA can be sensitive to the initial state of the feature extractor. If the feature extractor can not separate classes well, it can take a longer time to annotate the videos initially. After fine-tuning the network with new labels for a few epochs, the labeling time can reduce again. Besides, putting more video frames in the t-SNE plot can overflow the screen and make the annotation process harder for the oracle.

REFERENCES

[1] S. Poorgholi, O. S. Kayhan, and J. C. v. Gemert, *t-eva: Time-efficient t-sne video annotation,* in *International Conference on Pattern Recognition* (Springer, 2021) pp. 153–169.

[2] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, *Activitynet: A large-scale video benchmark for human activity understanding,* in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015) pp. 961–970.

[3] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, *The kinetics human action video dataset,* CoRR **abs/1705.06950** (2017), arXiv:1705.06950 .

[4] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, *Large-scale video classification with convolutional neural networks,* in *CVPR* (2014).

[5] K. Liu, W. Liu, C. Gan, M. Tan, and H. Ma, *T-c3d: Temporal convolutional 3d network for real-time action recognition,* in *AAAI* (2018).

[6] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, *C3D: generic features for video analysis,* CoRR **abs/1412.0767** (2014), arXiv:1412.0767 .

[7] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, *Temporal segment networks: Towards good practices for deep action recognition,* CoRR **abs/1608.00859** (2016), arXiv:1608.00859 .

[8] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, *A simple baseline for multi-object tracking,* (2020), arXiv:2004.01888 [cs.CV] .

[9] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, *Fast online object tracking and segmentation: A unifying approach,* CoRR **abs/1812.05050** (2018), arXiv:1812.05050 .

[10] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, *Distractor-aware siamese networks for visual object tracking,* CoRR **abs/1808.06048** (2018), arXiv:1808.06048 .

[11] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, and L. V. Gool, *Temporal 3d convnets: New architecture and transfer learning for video classification,* CoRR **abs/1711.08200** (2017), arXiv:1711.08200 .

[12] U. Ahsan, C. Sun, and I. A. Essa, *Discrimnet: Semi-supervised action recognition from videos using generative adversarial networks,* CoRR **abs/1801.07230** (2018), arXiv:1801.07230 .

[13] M. Zeng, T. Yu, X. Wang, L. T. Nguyen, O. J. Mengshoel, and I. Lane, *Semi-supervised convolutional neural networks for human activity recognition,* CoRR **abs/1801.07827** (2018), arXiv:1801.07827 .

[14] R. Girdhar, D. Tran, L. Torresani, and D. Ramanan, *Distinit: Learning video representations without a single labeled video,* CoRR **abs/1901.09244** (2019), arXiv:1901.09244 .

[15] D. Ghadiyaram, M. Feiszli, D. Tran, X. Yan, H. Wang, and D. Mahajan, *Large-scale weakly-supervised pre-training for video action recognition,* CoRR **abs/1905.00561** (2019), arXiv:1905.00561 .

[16] L. D. Alessandro Masullo, *Muvilab,* https://github.com/ale152/muvilab (2019).

[17] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, *Scaling egocentric vision: The EPIC-KITCHENS dataset,* CoRR **abs/1804.02748** (2018), arXiv:1804.02748 .

[18] A. Kläser, M. Marszalek, and C. Schmid, *A spatio-temporal descriptor based on 3d-gradients,* in *BMVC* (2008).

[19] P. Scovanner, S. Ali, and M. Shah, *A 3-dimensional sift descriptor and its application to action recognition,* in *Proceedings of the 15th ACM International Conference on Multimedia*, MM '07 (Association for Computing Machinery, New York, NY, USA, 2007) p. 357–360.

[20] L. Sevilla-Lara, Y. Liao, F. Güney, V. Jampani, A. Geiger, and M. J. Black, *On the integration of optical flow and action recognition,* CoRR **abs/1712.08416** (2017), arXiv:1712.08416 .

[21] H. Wang and C. Schmid, *Action recognition with improved trajectories,* in *2013 IEEE International Conference on Computer Vision* (2013) pp. 3551–3558.

[22] K. Simonyan and A. Zisserman, *Two-stream convolutional networks for action recognition in videos,* CoRR **abs/1406.2199** (2014), arXiv:1406.2199 .

[23] Z. Xu, Y. Yang, and A. G. Hauptmann, *A discriminative CNN video representation for event detection,* CoRR **abs/1411.4006** (2014), arXiv:1411.4006 .

[24] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell, *Actionvlad: Learning spatio-temporal aggregation for action classification,* in *Proceedings of (CVPR) Computer Vision and Pattern Recognition* (2017) pp. 3165 – 3174.

[25] Z. Shou, D. Wang, and S. Chang, *Action temporal localization in untrimmed videos via multi-stage cnns,* CoRR **abs/1601.02129** (2016), arXiv:1601.02129 .

[26] J. Carreira and A. Zisserman, *Quo vadis, action recognition? A new model and the kinetics dataset,* CoRR **abs/1705.07750** (2017), arXiv:1705.07750 .

[27] C. Feichtenhofer, H. Fan, J. Malik, and K. He, *Slowfast networks for video recognition,* in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2019).

[28] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, *Video action transformer network,* CoRR **abs/1812.02707** (2018), arXiv:1812.02707 .

[29] N. Hussein, E. Gavves, and A. W. M. Smeulders, *Timeception for complex action recognition,* CoRR **abs/1812.01289** (2018), arXiv:1812.01289 .

[30] B. Martinez, D. Modolo, Y. Xiong, and J. Tighe, *Action recognition with spatial-temporal discriminative filter banks,* (2019), arXiv:1908.07625 [cs.CV] .

[31] Z. Qiu, T. Yao, and T. Mei, *Learning spatio-temporal representation with pseudo-3d residual networks,* CoRR **abs/1711.10305** (2017), arXiv:1711.10305 .

[32] D. Tran, H. Wang, L. Torresani, and M. Feiszli, *Video classification with channel-separated convolutional networks,* CoRR **abs/1904.02811** (2019), arXiv:1904.02811 .

[33] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction* (Springer, 2007).

[34] A. Chatzimparmpas, R. M. Martins, and A. Kerren, *t-visne: Interactive assessment and interpretation of t-sne projections,* IEEE Transactions on Visualization and Computer Graphics **26**, 2696–2714 (2020).

[35] L. van der Maaten and G. Hinton, *Visualizing high-dimensional data using t-sne,* Journal of Machine Learning Research **9**, 2579 (2008), pagination: 27.

[36] G. C. Linderman and S. Steinerberger, *Clustering with t-sne, provably,* CoRR **abs/1706.02582** (2017), arXiv:1706.02582 .

[37] L. McInnes, J. Healy, and J. Melville, *Umap: Uniform manifold approximation and projection for dimension reduction,* (2018), arXiv:1802.03426 [stat.ML] .

[38] N. Pezzotti, J. Thijssen, A. Mordvintsev, T. Höllt, B. v. Lew, B. P. Lelieveldt, E. Eisemann, and A. Vilanova, *Gpgpu linear complexity t-sne optimization,* IEEE Transactions on Visualization and Computer Graphics (Proceedings of VAST 2019) **26** (2020).

[39] o. c. darrenl, *labelimg,* `https://github.com/tzutalin/labelImg` (2017).

[40] o. c. Amit K Gupta, *imglab,* `https://github.com/NaturalIntelligence/imglab` (2017).

[41] J. Li and J. Z. Wang, *Real-time computerized annotation of pictures,* IEEE Transactions on Pattern Analysis and Machine Intelligence **30**, 985 (2008).

[42] Gang Wang and D. Forsyth, *Joint learning of visual attributes, object classes and visual saliency,* in *2009 IEEE 12th International Conference on Computer Vision* (2009) pp. 537–544.

[43] J. Wu, Y. Yu, C. Huang, and K. Yu, *Deep multiple instance learning for image classification and auto-annotation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* (2015).

[44] F. P. S. Luus, N. Khan, and I. Akhalwaya, *Active learning with tensorboard projector,* CoRR **abs/1901.00675** (2019), arXiv:1901.00675 .

[45] A. Dutta and A. Zisserman, *The via annotation software for images, audio and video,* in *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19 (Association for Computing Machinery, New York, NY, USA, 2019) p. 2276–2279.

[46] L. van der Maaten, *Barnes-hut-sne,* (2013), arXiv:1301.3342 [cs.LG] .

[47] K. Hara, H. Kataoka, and Y. Satoh, *Learning spatio-temporal features with 3d residual networks for action recognition,* CoRR **abs/1708.07632** (2017), arXiv:1708.07632 .

[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, *Scikit-learn: Machine learning in Python,* Journal of Machine Learning Research **12**, 2825 (2011).

[49] M. Wattenberg, F. Viégas, and I. Johnson, *How to use t-sne effectively,* (2016).

7

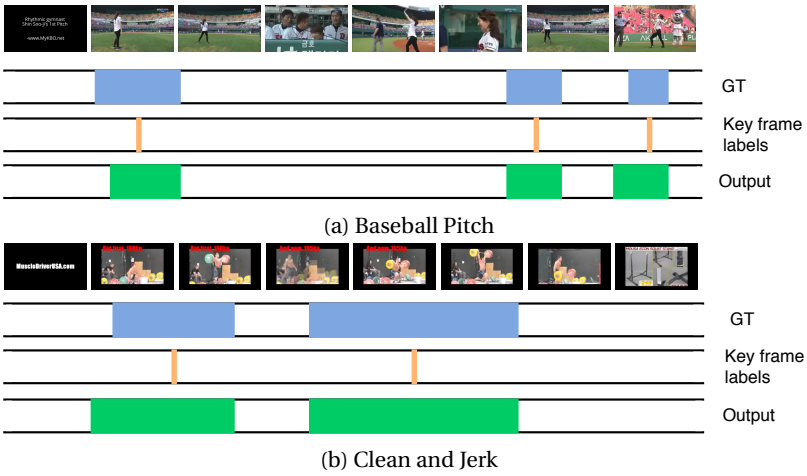
PUNet: Temporal Action Proposal Generation with Positive Unlabeled Learning using Key Frame Annotations

This chapter has been published as:
N. U. S. Zia, O. S. Kayhan, and J. van Gemert, Punet: Temporal action proposal generation with positive unlabeled learning using key frame annotations, in 2021 IEEE International Conference on Image Processing (ICIP) (2021) pp.2598–2602. [1]

ABSTRACT

Popular approaches to classifying action segments in long, realistic, untrimmed videos start with high quality action proposals. Current action proposal methods based on deep learning are trained on labeled video segments. Obtaining annotated segments for untrimmed videos is time consuming, expensive and error-prone as annotated temporal action boundaries are imprecise, subjective and inconsistent. By embracing this uncertainty we explore to significantly speed up temporal annotations by using just a single key frame label for each action instance instead of the inherently imprecise start and end frames. To tackle the class imbalance by using only a single frame, we evaluate an extremely simple Positive-Unlabeled algorithm (PU-learning). We demonstrate on THUMOS'14 and ActivityNet that using a single key frame label give good results while being significantly faster to annotate. In addition, we show that our simple method, PUNet[1], is data-efficient which further reduces the need for expensive annotations.

[1] https://github.com/NoorZia/punet



(a) Baseball Pitch

(b) Clean and Jerk

Figure 7.1: Our proposed method. A single frame is labeled for each action instance. The detected results are shown for THUMOS'14 dataset. Using a single frame, the PU learning network is able to detect action boundaries with low error.

7.1. INTRODUCTION

With videos naturally untrimmed and multiple actions per video, doing temporal action localization involves detecting all action labels, with their start and end time. Action localization methods [2, 3] utilize a two stage approach: 1. proposal generation and 2. action classification of each proposal.

Because proposal generation uses machine learning, it relies on annotated data. Such annotations for untrimmed videos have each action instance labeled with a start and end timestamp of the action and each video can have multiple, possibly overlapping, action instances [4]. Obtaining these labels is time consuming and expensive [5]. Moreover, the labeling of the action instances is subjective and error prone [6] due to a different understanding of action duration, thus affecting the results of the model trained using these labels [7]. Recent work in action recognition has shown that performance improves by using most discriminative portions of the video for training [8]. Similarly, work has been done to optimize the segment length and recognize human actions with fewer frames [9, 10]. Using a single timestamp instead of start and end time for action recognition has been shown to be a reasonable compromise between performance and annotation effort [11]. In this paper, we question the need for more complex methods, and evaluate an extremely simple idea: We propose labeling a single action frame as "key frame" inside an action's temporal window (Figure 7.1) and evaluate the simplest approach we could find: Positive Unlabeled (PU) learning to detect action frames.

Our approach requires a single labeled key frame belonging to the action instance. The remaining frames are a combination of background and unlabeled action frames, referred together as 'unlabeled data'. If we consider the unlabeled data as negative, the problem becomes imbalanced due to the high ratio of unlabeled data to positive which we tackle in a PU learning [12] setting where the true positives are iteratively removed

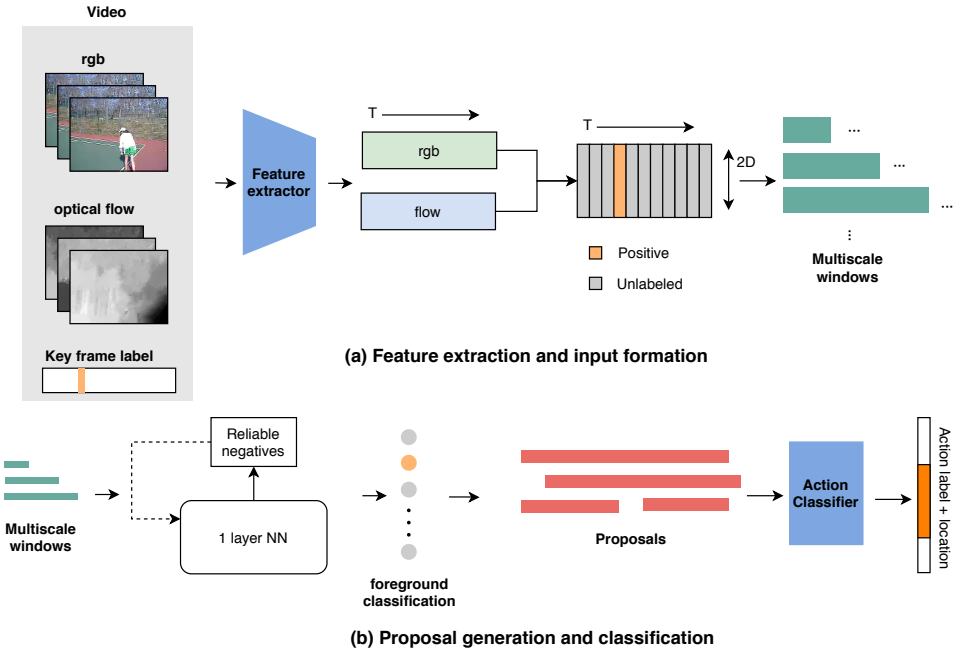


Figure 7.2: Overview. We use one labeled point for each action instance. The input is devided in non-overlapping windows for training using PU learning at different scales with I3D encoded features to extract proposals.

from the unlabeled data.

Our contributions are: 1) Instead of adding complexity, we evaluate the very simple Positive Unlabeled learning setting for action proposal generation using just a single labeled frame per action instance. 2) This simplistic method is able to achieve good results. 3) PU-learning is data-efficient: It does well when using a small number of action instances, allowing another reduction of the annotation effort.

Problem definition.

An untrimmed video sequence $X = \{x_n\}_{n=1}^{T}$ has T frames where x_n is the n-th frame in the video. Our single frame action annotations are $\Psi_g = \{\varphi_n = (t_{m,n})\}_{n=1}^{N_g}$ where $t_{m,n}$ is a selected frame at position m of the action instance n which we refer to as our key frame and N_g is the total number of action instances. For proposal generation, we have a binary action vs background classifier. We divide a video in non-overlapping windows, and a window is labeled positive if it contains a key frame.

7.2. METHOD

PU learning. We draw inspiration from the simple and elegant PU-learning algorithm [12] to train the binary action vs background classifier. It finds negative samples that are most dissimilar from the positives by refining such 'reliable negatives'. A Positive versus Unlabeled classifier is trained and tested on the unlabeled training set where high-

confidence predicted negative samples are deemed reliable negatives. The remaining unlabeled samples are removed from the training set. The size of the reliable negatives set is reduced iteratively by training a classifier using positive and reliable negative data and evaluating on reliable negative data points. Reliable negatives classified as positives are removed from the training set and this step is repeated until no positive classes are identified or the size of reliable negatives is less than positive samples. This step reduces the size of the negative samples and mitigates class imbalance.

Proposal generation and classification. The proposal generation module uses PU classifier to generate candidate proposals for each window scale. The results from different window scales are aggregated to get the final proposals. We use a state of the art action classifier [13] to classify our action proposals. The overview of PUNet can be seen in Figure 7.2.

7.3. EXPERIMENTS

Implementation details. We use I3D [14] pretrained on Kinetics [15] to extract RGB and optical flow features. The feature representations from RGB and optical flow are concatenated to obtain $(T \times 2D)$ features for a video of duration T. From untrimmed videos, we extract temporal windows of varying lengths, 16, 32, 48, 64, and 80 frames; with no overlap. For the proposal classifier, we use a single layer Multi-Layer Perceptron (MLP) with 100 hidden units. The single layer network is trained using adam optimizer and 10^{-4} learning rate. To extract the initial set of reliable negatives, the predicted negatives are thresholded based on their confidence score. The threshold value is set as 0.99.

Experimental Setup. We evaluate on THUMOS'14 [16], ActivityNet v1.2 and v1.3 [17] datasets. The THUMOS'14 dataset has temporal annotations for 20 classes with 200 training and 213 test videos. ActivityNet v1.2 has 100 action classes and 4,819 training, 2,383 validation and 2,480 test videos. ActivityNet v1.3 has 200 action classes, 10,024 training, 4,926 validation and 5,044 test videos. For ActivityNet, we use the validation videos for testing as the groundtruth for test videos is withheld. We measure performance with the F1-score. For temporal action proposal generation, the Average Recall (AR) as calculated at different IoU thresholds is used for evaluation. We also calculate AR with an average number of proposals (AR@AN) to determine relation between recall and number of proposals. For temporal action detection, mean average precision (mAP) is reported.

Results. A good proposal generation method should generate high recall with few proposals. PUNet compares well to most state of the art methods which use full supervision. We list the comparative results for THUMOS'14 in Table 7.1. We evaluate the quality of our generated proposals by comparing the recall at different tIoU thresholds (Figure 7.3). Our results have good recall at 100 proposals for tIoU 0.1 to 0.5. The results for action detection indicate that our extremely simple PUNet does well when compared to others. These results on THUMOS'14 are summarized in Table 7.2. Our method outperforms all weakly supervised methods except BaSNet [21], against which it shows a slight performance decrease while being more data efficient and having a simpler network design. Besides, our iterative approach takes around 4.6 minutes to train even on CPU. Our method can also be used to improve other single frame methods [11]. Compared to fully supervised methods, our method gives good performance while utilizing

Table 7.1: Comparison of our method with other state of the art proposal generation methods on THUMOS'14 dataset in terms of AR@AN. Our method outperforms all fully supervised methods at AR@50 and AR@100 except BSN.

Supervision	Method	@50	@100
Full	DAPs [18]	13.56	23.83
	Sparse [19]	13.42	21.44
	SST [20]	19.90	28.36
	TURN [2]	21.86	31.89
	BSN [3]	35.41	46.06
Weak	PUNet (ours)	32.72	40.61

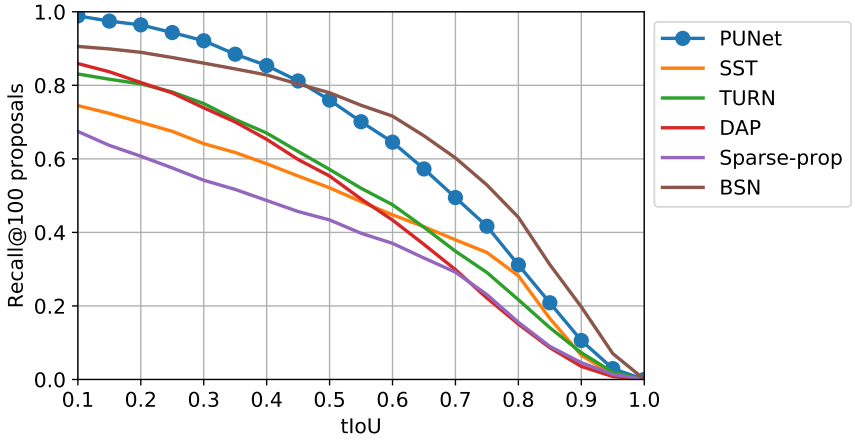


Figure 7.3: Comparison of our method with the state of the art fully supervised methods on THUMOS'14 dataset. Recall with 100 proposals at different tIoU thresholds show PUNet has high recall compared to all fully supervised methods when tIoU < 0.5. At higher tIoUs, PUNet outperforms all fully supervised methods except BSN.

significantly less annotation effort. Table 7.3 shows our results on ActivityNet v1.2 and v1.3. For ActivityNet v1.2, we see that our method outperforms all weakly supervised methods except BaSNet and is not too far behind the fully supervised method. On ActivityNet v1.3, our method outperforms all weakly supervised methods including BaSNet.

Qualitative analysis. The qualitative analysis of our approach for key frame annotation is shown in Figure 7.1. The GT denotes groundtruth segments and the labels denote the key frame inputs to our network. Without any postprocessing, our proposal evaluation model is able to capture the full extent of the temporal duration and not just the key frames.

Data efficiency. We evaluate how the performance of PUNet changes when trained with a small dataset. We train BaSNet and PUNet with various training set sizes of THUMOS'14 dataset and report the mean average precision. All classes are included in each training set in an equal ratio.

Table 7.2: Comparison of our method with the state of the art methods on the THUMOS'14 dataset. Average mAP is reported at IoU thresholds from 0.1 to 0.5. Weak * indicate use of additional information in weakly supervised approach. PUNet outperforms most weakly supervised and some fully supervised methods while utilizing less annotations.

Supervision	**Method**	**AVG mAP**
Full	Yuan et al. [22]	35.7
	TAL-Net [23]	52.3
	P-GCN [24]	61.6
Weak (video)	UntrimmedNet [13]	29.0
	Liu et al. [25]	40.9
	BaSNet [21]	43.6
Weak (single frame)*	SF-Net [11]	51.5
	PUNet (ours)	42.1
	SF-Net + PUNet (ours)	53.6

Table 7.3: Comparison on ActivityNet (Anet) v1.2 and v1.3 with the current state of the art methods. PUNet has comparable performance to fully supervised method and outperforms most weakly supervised methods for action localization.

Supervision	**Method**	**AVG mAP**	
		ANet v1.2	ANet v1.3
Full	S-CNN [26]	26.6	-
	CDC [27]	-	23.8
Weak	Liu et al. [25]	22.4	21.2
	BaSNet [21]	24.3	22.2
Weak (single frame)*	PUNet (ours)	23.7	22.5

Results are shown in Figure 7.4. For small training sets, PUNet outperforms BaSNet. As the data size increases, the performance becomes more similar for both. With 20 training samples, PUNet achieves 14.7% performance gain. The performance gain reduces as the training data set size increases.

Generalizability of proposals. We evaluate the generalization ability of PUNet by testing its performance on unseen action classes. We randomly leave one, two and three classes from our training set and test on our test set containing all 20 classes of THUMOS'14 data. As shown in Table 7.4, there is only a slight performance decrease when testing on unseen classes and the method is able to generate high quality proposals on unseen classes.

Annotation speed for different settings. Annotation time required to label a single key frame and the full segment is measured for some videos from THUMOS'14 dataset. Five videos are selected from THUMOS'14 dataset with different classes and six annotators are chosen. Three annotators are asked to label the full segment and the remain-

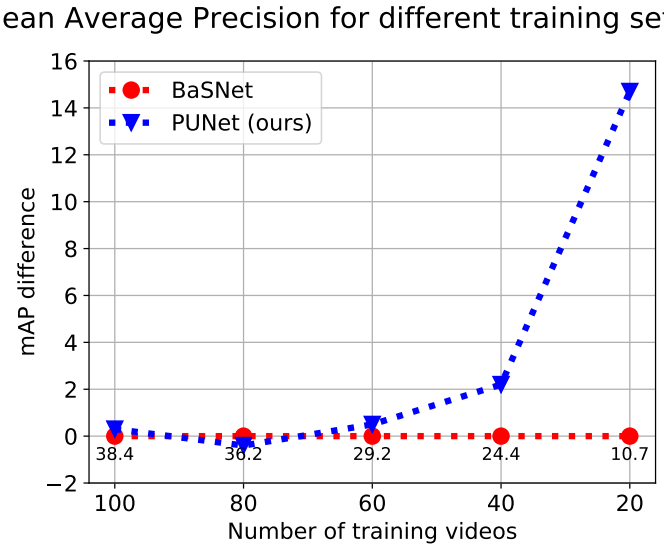


Figure 7.4: Data Efficiency. We compare the performance of BaSNet and PUNet when training data is reduced per class from 1 to 5 videos. For small training set, PUNet has a higher relative performance. The performance becomes similar when training set size increases.

Table 7.4: Generalization evaluation of PUNet on THUMOS'14 dataset. Action classes are removed from the training set and the resulting model is evaluated on the full test set (seen + unseen classes) containing 20 classes.

# classes in training set	AR@50	AR@100
17	31.8	38.5
18	32.4	39.3
19	32.5	40.2
20	32.7	40.6

ing three are asked to label a single frame for every action occurrence. On average, one minute video takes 65 seconds for single frame labeling and 250 seconds for full segment labeling.

7.3.1. How many annotations per video are actually needed?

Videos in THUMOS'14 have 15 action instances on average which are spread unevenly among the videos with a standard deviation of 24, and range from 1 to 128 per video. The total labeled action instances in the training set are shown in Figure 7.5. We evaluate whether annotations for all instances are needed to get an effective action proposal network. F1-score is used to compare the maximum annotations per video ranging from 1 to 128.

After a maximum limit of 6 annotations per video, F1-score has low variance (Figure 7.5). PUNet is able to identify the unlabeled key frames effectively. Results indicate that

Table 7.5: Effect of using limited annotations on action localization for THUMOS'14 dataset. We set the maximum annotations per video to 6 to train these models. The action instances needed reduce by one-third from 3007 to 947. The performance only decreases slightly for weakly supervised methods and increases by 0.9% for fully supervised method.

Supervision	Method	Whole	Partial
Full	GTAD [28]	55.4	56.3
Weak	BaSNet [21]	43.6	42.1
Weak*	PUNet (Ours)	42.1	41.3

not all annotations are necessary to achieve a good performance.

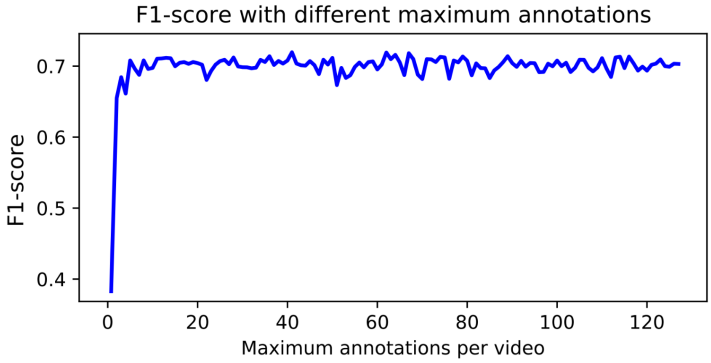


Figure 7.5: Effect of changing the maximum number of annotations per video on the binary classifier performance. After 6 annotations per video, the performance does not change much and the standard deviation reduces. The mean value of F1-score from 1-128 annotations is 0.69 ± 0.05, and mean F1-score from 6-128 annotaions is 0.70 ± 0.008. Our method can achieve good results without using all the annotations.

In Table 7.5, we show that not all action annotations are required for good detection performance by training fully and weakly supervised action localization networks. Thus, we set the number of maximum annotations per video to 6 action instances. The number of action instances reduces from 3007 to 947. We train PUNet with a maximum of 6 annotations per video and obtain a slight performance drop of 0.8%. Similarly, BaSNet [21] is trained with the reduced video size and the results show a 1.5% reduction in mAP. Fully supervised method, GTAD [28], is trained with only six labeled action instances and the rest of the data is unlabeled. Interestingly, the performance increases by 0.9%. The results indicate that the methods do not need all the labels to obtain good results.

7.4. CONCLUSION

We use key frame level supervision for training temporal action proposal model in a PU-learning algorithm on three untrimmed datasets. Compared to fully supervised methods and other weakly supervised methods, this extremely simple approach generates proposals with high recall and high temporal overlap. Experimental evaluation

on THUMOS'14 shows that: (i) Using a key frame annotation gives comparable performance to using fully supervised annotation which uses start and end annotations, (ii) All action instances from one video are not necessary to achieve good detection results, (iii) Our results are comparable to the state of the art methods and data efficient. We conclude that annotation effort can be significantly reduced by labeling key frames and for long untrimmed videos, only a limited number of action instances need to be labeled and trained to achieve similar results.

REFERENCES

[1] N. U. S. Zia, O. S. Kayhan, and J. v. Gemert, *Punet: Temporal action proposal generation with positive unlabeled learning using key frame annotations,* in *2021 IEEE International Conference on Image Processing (ICIP)* (2021) pp. 2598–2602.

[2] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia, *Turn tap: Temporal unit regression network for temporal action proposals,* in *ICCV* (2017).

[3] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, *Bsn: Boundary sensitive network for temporal action proposal generation,* in *ECCV* (2018).

[4] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei, *Every moment counts: Dense detailed labeling of actions in complex videos,* IJCV (2017).

[5] S. Poorgholi, O. S. Kayhan, and J. C. van Gemert, *t-eva: Time-efficient t-sne video annotation,* arXiv preprint arXiv:2011.13202 (2020).

[6] D. Moltisanti, M. Wray, W. Mayol-Cuevas, and D. Damen, *Trespassing the boundaries: Labeling temporal bounds for object interactions in egocentric video,* in *ICCV* (2017).

[7] S. Satkin and M. Hebert, *Modeling the temporal extent of actions,* in *ECCV* (2010).

[8] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce, *Automatic annotation of human actions in video,* in *ICCV* (2009).

[9] K. Schindler and L. Van Gool, *Action snippets: How many frames does human action recognition require?* in *CVPR* (2008).

[10] X. Yang and Y. Tian, *Effective 3d action recognition using eigenjoints,* Journal of Visual Communication and Image Representation (2014).

[11] F. Ma, L. Zhu, Y. Yang, S. Zha, G. Kundu, M. Feiszli, and Z. Shou, *Sf-net: Single-frame supervision for temporal action localization,* ECCV (2020).

[12] D. H. Fusilier, M. Montes-y Gómez, P. Rosso, and R. G. Cabrera, *Detecting positive and negative deceptive opinions using pu-learning,* Information processing & management (2015).

[13] L. Wang, Y. Xiong, D. Lin, and L. Van Gool, *Untrimmednets for weakly supervised action recognition and detection,* in *CVPR* (2017).

[14] J. Carreira and A. Zisserman, *Quo vadis, action recognition? a new model and the kinetics dataset,* in *CVPR* (2017).

[15] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, *et al.*, *The kinetics human action video dataset,* arXiv preprint arXiv:1705.06950 (2017).

[16] Y.-G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, *Thumos challenge: Action recognition with a large number of classes,* (2014).

[17] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, *Activitynet: A large-scale video benchmark for human activity understanding,* in *CVPR* (2015) pp. 961–970.

[18] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem, *Daps: Deep action proposals for action understanding,* in *ECCV* (2016).

[19] F. Caba Heilbron, J. Carlos Niebles, and B. Ghanem, *Fast temporal activity proposals for efficient detection of human actions in untrimmed videos,* in *CVPR* (2016).

[20] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. Carlos Niebles, *Sst: Single-stream temporal action proposals,* in *CVPR* (2017).

[21] P. Lee, Y. Uh, and H. Byun, *Background suppression network for weakly-supervised temporal action localization.* in *AAAI* (2020).

[22] Z. Yuan, J. C. Stroud, T. Lu, and J. Deng, *Temporal action localization by structured maximal sums,* in *CVPR* (2017).

[23] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, *Rethinking the faster r-cnn architecture for temporal action localization,* in *CVPR* (2018).

[24] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, *Graph convolutional networks for temporal action localization,* in *ICCV* (2019).

[25] D. Liu, T. Jiang, and Y. Wang, *Completeness modeling and context separation for weakly supervised temporal action localization,* in *CVPR* (2019).

[26] Z. Shou, D. Wang, and S.-F. Chang, *Temporal action localization in untrimmed videos via multi-stage cnns,* in *CVPR* (2016) pp. 1049–1058.

[27] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, *Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos,* in *CVPR* (2017).

[28] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem, *G-tad: Sub-graph localization for temporal action detection,* in *CVPR* (2020).

8

CONCLUSION

In this thesis, we investigate location and temporal information with CNNs. The individual chapters can be summarized as follows.

We investigate how absolute location is encoded in CNNs by exploiting image border effects. We evaluate various convolution types in terms of their border handling, where some methods break translation equivariance and therefore allow CNNs to exploit the absolute spatial location. Recovering translation equivariance improves robustness to translation and increases data efficiency.

We also investigates object locality by analyzing three different deep object detectors and their object-context relationships. The results show that single-stage and two-stage object detectors can access and use the context depending on the size of their receptive field. Context helps for most of the classes and eases the learning process. In contrast, context hurts accuracy in the particular application of visually verifying if an object is present in an image. To investigate visual verification, we introduce the Delft Bikes dataset that includes 22 object parts of a bike with their location and state labels. The analyses indicate that object detectors hallucinate bike parts with high overlap with a possible correct position when the part is not visible in the image. Furthermore, the thesis studies the effect of data augmentation on automatically human pose estimation, specifically investigating occlusions. We design occlusion attacks to measure the robustness of the current models and propose occlusion-based data augmentation techniques. The results show that current pose estimators are sensitive to occlusion, and data augmentation does not bring sufficient solution to the occlusion problems.

In addition to spatial localization, the thesis analyzes temporal localization in video. We investigate time-efficient spatio-temporal video labelling by mapping frames to 2D, where similar frames are mapped to similar 2D locations, which allows easy grouping and offers a 6 to 12 times faster labeling. We further investigate how to reduce the temporal labeling effort for temporal action localization tasks. We propose a weakly-supervised Positive-Unlabeled learning approach that uses only single frame labels and yields a label efficient solution.

8

THE LOCATION BIAS DILEMMA

Exploiting location information for automatic image and video analysis generally provides benefits (Chapter 2, 3, 5, 6 and 7) resulting in better accuracy; however, at the same time, location sometimes results in detrimental outcomes (Chapter 2, 4 and 5). For example, object detectors benefit from location biases (Chapter 3), yet, these very same biases can also cause hallucination problems in visual verification (Chapter 4). Also for videos, 3D CNNs can exploit the spatial context. As shown in Chapter 6, we can use a trained 3D CNN backbone to obtain the embedding of video clips. When projecting the video features on 2D space, the videos with similar context or background are often placed closer to each other. If the action classes of the videos are similar, spatial context provides useful solutions. On the other hand, if the different actions happen in a similar place, for instance, doing two unrelated actions in the same kitchen, then the 2D projection may place semantically different video segments close to each other. Therefore, the contribution of location biases depends on the tasks and models. This thesis makes such location biases explicit, and thus raises awareness that location bias is an important factor to take into account in automatic image and video analysis.

EQUIVARIANT ARCHITECTURES VS DATA AUGMENTATION

Two chapters in this thesis each explore a different strategy for neural networks to deal with transformations that should not change the final classification output. One strategy is data augmentation (Chapter 5) where the existing data is augmented by adding transformations of existing images to the training data. The other strategy is hard coding equivariance and invariance to these transformations in the neural network architecture itself (Chapter 2). These two strategies have the same goal, yet their differences make them well-suited in different scenarios. The strategy of hard-coding equivariance in the network architecture benefits in scenarios where there are a small number of transformations which are known to occur often in the data [1–7]. In contrast, data augmentation cannot give hard guarantees as it relies on stochastic training yet it easily allows for multiple transformations to occur at the same time [8, 9]. Interestingly, in this thesis, we found that data-augmentation has difficulty when dealing with occlusions (Chapter 5) and a neural network that encodes shift-equivariance will do its very best to still exploit location information for cases where location information benefits the learning objective. It would be interesting to do further research to explicitly contrast data-augmentation with equivariance, investigate their differences, and determine in which cases equivariance is preferred, and in which cases data-augmentation.

GOING BEYOND THE BOUNDARIES

Chapter 2 shows that hard-coding position equivariance in the architecture can and will let the network exploit border effects to make use of position information for image classification. It would be interesting to evaluate border effects for other tasks explored in other chapters of the thesis, such as video analysis, object detection, and pose estimation. What is more, other padding values and methods such as replication and reflection padding might play another role in allowing the network to exploit position information. Moving beyond border effects in position equivariance, there might be other short-cuts that a network can exploit to break equivariance for other types of transformations such as rotation [1, 2, 6, 10], mirroring [5, 10], and scaling [3, 4]. A generic testing framework to evaluate the empirical equivariance of network architectures in a plug and play manner would be a useful tool for the community.

HELPING HUMANS PERFORM VISUAL VERIFICATION

The new visual verification task proposed in chapter 4 is a specific use case for object detection: detect if a common object is missing from its usual location. One direction of follow-up research is an evaluation if the tight bounding box outcome as preferred by an object detector matches with how humans see a detection, as it might be more informative to show a bit more of the context around an object. Other extensions include collecting larger and more varied datasets, and moving the visual verification task to other domains such as 3D and video.

REFERENCES

[1] M. Weiler, F. A. Hamprecht, and M. Storath, *Learning steerable filters for rotation equivariant cnns,* in *Proceedings of the IEEE Conference on Computer Vision and*

Pattern Recognition (2018) pp. 849–858.

[2] T. Cohen, M. Geiger, and M. Weiler, *A general theory of equivariant cnns on homogeneous spaces,* arXiv preprint arXiv:1811.02017 (2018).

[3] I. Sosnovik, M. Szmaja, and A. Smeulders, *Scale-equivariant steerable networks,* arXiv preprint arXiv:1910.11093 (2019).

[4] D. E. Worrall and M. Welling, *Deep scale-spaces: Equivariance over scale,* arXiv preprint arXiv:1905.11697 (2019).

[5] M. Weiler and G. Cesa, *General e(2)-equivariant steerable cnns,* arXiv preprint arXiv:1911.08251 (2019).

[6] D. Romero, E. Bekkers, J. Tomczak, and M. Hoogendoorn, *Attentive group equivariant convolutional networks,* in *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 119, edited by H. D. III and A. Singh (PMLR, 2020) pp. 8188–8199.

[7] M. J. Hutchinson, C. L. Lan, S. Zaidi, E. Dupont, Y. W. Teh, and H. Kim, *Lietransformer: Equivariant self-attention for lie groups,* in *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 139, edited by M. Meila and T. Zhang (PMLR, 2021) pp. 4533–4543.

[8] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, *Cutmix: Regularization strategy to train strong classifiers with localizable features,* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019) pp. 6023–6032.

[9] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, *Augmix: A simple data processing method to improve robustness and uncertainty,* arXiv preprint arXiv:1912.02781 (2019).

[10] T. Cohen and M. Welling, *Group equivariant convolutional networks,* in *International conference on machine learning* (PMLR, 2016) pp. 2990–2999.

ACKNOWLEDGEMENTS

Time flies, I still remember the day I came to Delft. These four years seem to me like four months. When I look back, I see that I obtained great friendships, memories, successes and happiness in this seemingly short time. This journey is not just an academic adventure; it is a state of witnessing an important moment in my life. The sum of the victories won against all adversities despite the flowing river of time...

First of all, I would like to express my sincere gratitude to Marcel, more than a promoter who has never spared his support from the very beginning and always helps in every single situation. I also would like to express my endless gratitude to my copromoter Jan, who has always been a guide and role model from the origin of this path and acted as a friend beyond a supervisor. He is always a good listener in all matters and tolerates my conversational dynamicity.

I met people from many different groups, nationalities, backgrounds, languages and perspectives. All these good people became friends and companions on this journey. I might not have gone this far without them, just as Frodo could not do without Sam on the way to Mount Doom.

First, I'll start with my group, Computer Vision. I would like to thank; Silvia besides her friendship, for her kind attitude in my PhD interview, then sharing the same room with me for four years and her contributions to turn our last room into a botanic garden; Seyran, who started at the same time as me, for often bringing another perspective to the problems with her signal processing background and delicious saffron dessert; Wenjie, who is actually from PR yet I can count him in CV, my first roommate, a mechanical keyboard collector who got me used to pytorch, and my next door neighbour before he left; Yancong, for being such an honest and great friend who can learn things in a fast manner like his football skills; Yeshwanth, who could not recognise me with short hair, for his genial nature and cooking the best butter chicken for us; Ziqi, who is the craziest and most elegant Chinese and loves Korean culture, for pleasant conversations; Attila and Robert-Jan, who brought a new atmosphere to the group, carried our relations to the next level with activities such as workshops and reading groups that we organized together, for your friendship and having a chance to speak Dutch with you; Nergis for countless time turning any topic into an interesting conversation that contains a series of independent topics; Xin, for her hardworking and calm attitude; Ombretta, with whom I have enjoyed working and talking, for always showing Italian hospitality; Amogh, Marian and Nikolaas, unfortunately, I could not spend much time because they are half industry guys, for having fun together when we see each other; Yunqiang and Burak, who possess a quiet nature, for their benevolence; Abolfazl, who I become a close friend very quickly and still keep in touch, for accompanying me with foosball matches; Marcos, for being my Spanish brother, having a lot in common, and understanding each other very well; Reza, for exciting endless conversation in which it is hard to convince him in many cases.

I continue to thank the Bioinformatics Group, which I always listen to their conversations and try to learn their topics, even if I sometimes do not have a full grasp of what they do. I would like to start with Stavros and Tamim, with whom we have common cultural richness and similarities. Our very first conversation was "How do you cook *musakka* (moussaka) ?". We organised football and basketball events numerous times together and they discovered my little talent in basketball. Ramin, more than a friend, never gets tired to explain to me Bioinformatics and never rejects me when I ask him to drink tea or coffee. I had an opportunity to work on protein localisation with Stavros and Soufiane, who is the most unaccented French I know. We regret stopping to work on our idea which became a Nature paper recently. Alex (aka Sally) is a real sportsman, no one knows how many sports he has tried. Thomas is knowledgeable about everything and always reminds me of Chandler in Friends owing to his jokes and joking style. Ahmed is always elegant and hospitable. Joana has a friendly and calm manner. Arlin has energetic and cheerful nature with social interests and hobbies like playing a guitar, kickboxing. Christine adds positive energy to the group and is always hungry. Christian always has an interesting topic to entertain us during Thursday borrels. Amelia brought a good vibe to the group even if she was with us for a short period. I also thank Tom, Aysun, Lieke, Mostafa, Mohammed, Eric, Meng, Thies and Amin, who provided good conversations at various times and places.

I would like to thank the great members of the PR group; Marco, David, for inspiring me a lot with their perspective on science, I listened to their lectures many times without getting bored, and I learned how to convey knowledge better; Bob for teaching me first Dutch words and opening new horizons in every his speech; Tom for being a good friend with his noisy laughs sharing many great moments, retreat 2017, DL course 2018, organising Maastricht retreat 2019 and so on; Alexander, who applied his theoretical mathematics knowledge in bodybuilding and achieved incredible results, for being my teammate in Central Powers during laser tag game; Arman, who always calls me "Abi", for talking about all kinds of subjects in our conversation that always starts with a cup of tea; Ramin, for spicing up our tea conversations with Arman; Wouter for his unlimited enthusiasm for learning and on-the-spot questioning skills; Jesse for inspiring me with his excellent presentation techniques; Yazhou and Jin, always a bit quiet and yet friendly; Lorenzo, for coming out with me to get some fresh air; Mahdi, for good conversations and practical advice such as where to run if you see a bear on a mountain.

I also thank people from Socially Perceptive Computing Lab (I checked the website for the correct name) aka *Social blabla*. I always enjoy talking to Hayley due to both the content and her lovely British accent. She is always a good listener and addresses different topics in her presentations. Honestly, after Ekin left the group, our corridors became silent. He always brings a cool vibe to Thursday borrels and other social activities. He also showed magical skills in the football field after ten years of break, yet his new career ended with a broken finger. Laura is not only Ekin's twin sister but also a sister to everyone in the group. She is always kind and helpful. Janxia is positive, open-minded and talkative. Chirag is one of the rare people who really represents the name of his group. After all years, he never loses his passion for football and scores many goals. However, I am not sure if his mind and feet agree on the things he wants to do on the field. Jose is often silent, yet he talks only when it matters. He is also a responsible person who arranges lab

and coffee talk schedules. Stefanie is another silent but friendly person with who I had nice conversations. Bernd gave me advice on learning Dutch and fruitful conversations about context.

I feel lucky to have Saskia, Ruud and Bart in our group, who always help, solve problems countless times, namely lifesavers and made other groups say "Problems are always handled on the 5th floor.". I would also like to thank Münire, Paulo and Ayşe Abla for giving me the opportunity to have a warm conversation every time we meet, no matter what the language is.

In addition, I would like to thank people I know not only within the PRB but also from other groups: Zeki for his humorous personality, sometimes it is not easy to understand if it is a joke or not; Chibuke, for being a real Nigerian striker and goal machine, unfortunately not seeing his prime time; Gamze, for her modesty and making delicious Turkish coffee; Miray, for being talkative. I would like to thank Manel, Xiuxiu, Zhe, Jaehun, Elvin, Alberto and Omar from the MMC group, whom we talked numerous times about PhD, research and life. We had the opportunity to get to know different cultures, arranged joint BBQs and made office life more fun. Besides, I thank Manel for sharing eid traditions and sweeties with us, Amirmasoud for our conversations about research, music and football, and Paul for organizing Deep Learning workshops together.

I extend thanks to Eric, Aladdin, Mateusz, Lima, Mauricio, Alberto, Sobhan, Niccolo, Calm, Ruben, Rafael, Irene and everyone who participated in the football and basketball activities we have organized.

I would also like to thank new PRB members Hadi, Rikard, Yasin, Skander, Xucong, with whom I had little opportunity to get to know but always enjoyed talking.

My PhD adventure did not pass by only doing research. I also found the opportunity to develop myself as a supervisor and lecturer. I would like to thank my master students Rafał, Soorosh, Andrei, Noor, and Vanathi, and my bachelor students Matthijs, Iwan, Matthijs and Max. I would also appreciate Yadong, Kanav, Dhruv, Chinmay, Bianca, Ekaterina, Leonid, Noor, Navin, Yapkan, Pranjal, who assisted in the Deep learning course for years, for their help and effort.

I express my gratitude to Prof. dr. Mehmet Korürek, Prof. dr. Avni Morgül, Prof. dr. Bahattin Karagözoğlu, Prof. Dr. Ali Y. Çamurcu, Prof. dr. Fevzi Yılmaz, Dr. İsa Yıldırım, Prof. dr. Ender M. Ekşioğlu, Dr. Orhan Özhan, Dr. Sadullah Öztürk and Dr. Cenk Aksoylar and my all other teachers who support me.

I should mention my Turkish friends who helped me to overcome my longing and home-sickness and created such a joyful environment whilst I have been in the Netherlands. Hamdi was so helpful since the first day eased my adaptation process. He was not just a friend, more of an older brother to me. I believe that we have an everlasting friendship. Oğuzhan, who started his PhD almost the same day as me, is the default host for our meeting. He is a great and strong man who can handle many difficulties and jokes. He still needs to improve his pop culture and comedy knowledge to understand me fully. If I need to define Taygun with one word: wise. Not only with his interest in science but also with his interest and knowledge on numerous subjects. He is a complete bookworm: both for reading and collecting. I also thank: Kasım Sinan for his modesty and joyful conversations; Baran for his logical thoughts and arguments; Ahmet Faruk for being my only friend here from ITU and his basketball skills; Hale for finding practical

and low budged solutions; Cezmi Bey for helping me regardless of what and where.

Even though we are in different countries, I would like to thank my long-lasting friends who always supported me. Huseyin, who is more than a friend, has always been with me since high school. We spent wonderful times in various places. He visited me in Delft with his lovely wife Sevdan. My relationship with Sefa and Oğuzhan just proves that distances are just a number. I can talk about all kinds of topics and issues with them for hours without getting bored. They always accept and understand me. We never had any disagreement or conflict. Orhan was my housemate before I moved to the Netherlands and gave me the privilege to become his witness at his wedding. I met Sinan during the MSc interview. Afterwards, life brought us together as research assistants at the same university and made us inseparable friends who can understand each other fully. Mahmud has been an older brother to me since the day I met him. He is a great person and supporter. I always enjoy fruitful conversations with Vahit including Beşiktaş. In addition, I want to thank my friends Ahmet, Cağrı, Osman, Hakan, Hasan, Furkan, Okan with who I still kept in touch since our bachelor studies.

Last but the most, I would like to express my biggest gratitude to my mother and father who have committed their life to me and my brothers and tried to raise us as good people. They always believe in me and support me in all circumstances. I always feel lucky to be raised in a big family where I was never alone, felt loved and cared for. Therefore, I appreciate my brothers, grandparents, aunts, uncles and cousins for their never-ending support and relationships. I would like to express my endless gratitude to my girlfriend, who has been on this journey with me since the beginning, for showing all support, commitments, sacrifices and care.

All the good people whose names I have mentioned and forgotten. I am so glad to have you all! This thesis is not an end, but rather it is a door to a larger room. I want to complete my words with the following verse of Yunus Emre:

"İlim ilim bilmektir.
İlim kendin bilmektir.
Sen kendini bilmezsin,
Ya nice okumaktır?"

Osman Semih Kayhan
2022 January, Delft.

CURRICULUM VITÆ

Osman Semih KAYHAN

07-02-1988 Born in Afyonkarahisar, Turkey.

EDUCATION

2008–2013 BSc in Electronics Engineering
Istanbul Technical University
Thesis: Mobile Arrhythmia Device Design
Promotor: Prof. dr. M. Korürek

2013–2016 MSc in Biomedical Engineering
Istanbul Technical University
Thesis: Automatic Rapid Diagnostic Test Reader Platform
Promotor: Dr. H. Özkan, Dr. İ. Yıldırım

2017–2021 PhD Computer Science
Computer Vision Lab, Delft University of Technology
Thesis: Thesis: Locality in Space and Time for Data-Efficient Visual Recognition
Promotor: Prof. dr. ir. M. J. T. Reinders
Copromotor: Dr. J. C. van Gemert

WORK EXPERIENCES

2014 Project Engineer, The Emerging Circuits and Computations Group
Istanbul Technical University

2014–2017 Research Assistant
Fatih Sultan Mehmet Vakıf University

2021 Deep Learning Specialist
Bosch Security Systems B.V.

AWARDS

2008 Ranked 0.2% at Turkish University Examination over 1.5M students.

2020 Outstanding Reviewer at British Machine Vision Conference

2020 Best project at Vision Understanding and Machine Intelligence challenge

LIST OF PUBLICATIONS

Thesis Research:

- **O. S. Kayhan** and J. C. van Gemert, *On translation invariance in cnns: Convolutional layers can exploit absolute spatial location*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020). (Chapter-2)
- **O. S. Kayhan** and J. C. van Gemert, *Evaluating the context for deep object detectors*, arXiv e-prints (2021). (Chapter-3)
- **O. S. Kayhan**, B. Vredebregt, and J. C. van Gemert, *Hallucination in object detection — a study in visual part verification*, in 2021 IEEE International Conference on Image Processing (ICIP) (2021) pp. 2234–2238. (Chapter-4)
- R. Pytel, **O. S. Kayhan**, and J. C. van Gemert, *Tilting at windmills: Data augmentation for deep pose estimation does not help with occlusions*, in 2020 25th International Conference on Pattern Recognition (ICPR) (2021) pp. 10568–10575. (Chapter-5)
- S. Poorgholi, **O. S. Kayhan**, and J. C. van Gemert, *t-eva: Time-efficient t-sne video annotation*, in International Conference on Pattern Recognition (Springer, 2021) pp. 153–169. (Chapter-6)
- N. U. S. Zia, **O. S. Kayhan**, and J. van Gemert, *Punet: Temporal action proposal generation with positive unlabeled learning using key frame annotations*, in 2021 IEEE International Conference on Image Processing (ICIP) (2021) pp.2598–2602. (Chapter-7)

Other Research:

- A. Lengyel, R-J. Bruintjes, M. B. Rios, **O. S. Kayhan**, D. Zambrano, N. Tomen and J. C. van Gemert, *VIPriors 2: Visual Inductive Priors for Data-Efficient Deep Learning Challenges.*, arXiv preprint arXiv:2201.08625 (2022).
- R-J. Bruintjes, A. Lengyel, M. B. Rios, **O. S. Kayhan**, and J. C. van Gemert, *VIPriors 1: Visual Inductive Priors for Data-Efficient Deep Learning Challenges.*, arXiv preprint arXiv:2103.03768 (2021).
- H. Özkan, and **O. S. Kayhan**, *A novel automatic rapid diagnostic test reader platform*, Computational and Mathematical Methods in Medicine 2016.
- H. Özkan, and **O. S. Kayhan**, *Automatic reading of helicobacter pylori tests via tablet computer*, National Conference on Electrical, Electronics and Biomedical Engineering. IEEE, 2016.
- H. Özkan, and **O. S. Kayhan**, *Evaluation of Adeno-Rota virus tests via tablet computer*, Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT). IEEE, 2016.
- A. Ç. Bağbaba, B. Örs, **O. S. Kayhan**, and A. T. Erozan, *JPEG image Encryption via TEA algorithm*, in 2015 23nd Signal Processing and Communications Applications Conference (SIU) (pp. 2090-2093), (2015, May), IEEE.