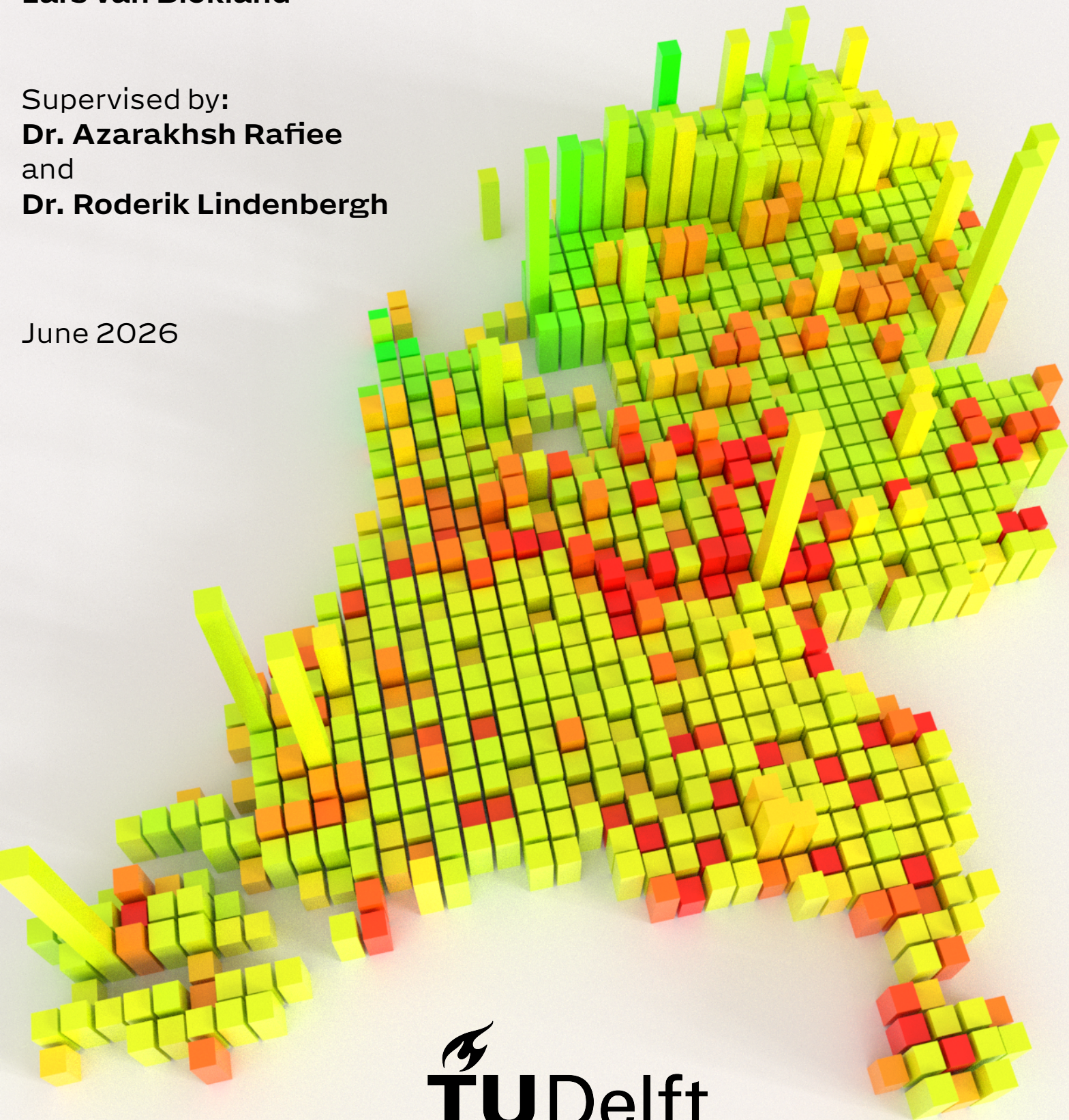


Air Temperature Estimation Through Thermal Satellite Imagery Using Uncertainty Integrated Transformers

A masters thesis by:
Lars van Blokland

Supervised by:
Dr. Azarakhsh Rafiee
and
Dr. Roderik Lindenbergh

June 2026



Air temperature estimation through thermal satellite imagery using uncertainty-integrated transformers

Lars van Blokland¹

¹ Technical University of Delft – Delft, the Netherlands

Keywords: Remote sensing, Deep learning, Spatio-temporal transformer, Uncertainty integrated.

This thesis is split into six parts:

1. A general introduction of the thesis topic
2. An explanation of concepts utilized in part 3
3. A scientific article detailing the research and results
4. A closing discussion
5. An overview of the projects reproducibility
6. A statement on AI usage

References for part 3 are included at the end of that section. References for all other parts are visible after part 6.

1. General introduction

This thesis details the design and testing of a machine learning model. The function of this model is to predict the temperature of the air at specific points in time and space.

Machine learning models train on data that is accompanied by a ground truth value. The model processes the input data, produces an estimate for the desired output, and this estimate is then compared to the ground truth. Using this comparison, the parameters that comprise the model are adjusted. Through repeated iterations of this process, the adjustments made cause the model to “learn” how to translate its input data into approximations of their ground truth. In the context of this research these input data are satellite images of land surface temperature (LST), and the ground truth are air temperature readings made by measuring stations.

LST and air temperature have a complex and correlated relationship (Kawashima et al., 2000). As the sun heats up the ground, a part of this warmth is radiated back into the air. This process is affected by surface topology, vegetation, exposure to sunlight, material of the ground, wind speed and other factors. For this reason estimating air temperature requires more than just LST data. Conveniently, many of these ancillary data types can also be derived using satellite imagery. The benefit of utilizing remotely sensed information to make predictions for the air temperature is primarily the massive coverage they provide.

A critical element of determining the efficacy of greater coverage is knowing the resolution of that coverage. The benefits of having data available for a large area are nullified if said data cannot be used to discern important details and variations. For that reason this thesis makes use of data captured by NASA's ECOSTRESS mission (NASA, 2018). ECOSTRESS is a relatively new instrument attached to the ISS and it provides LST data at a 70 by 70 meter pixel size. Many previous studies utilizing LST data used imagery provided by MODIS (Salomonson et al., 1989), which has a maximum resolution of 250 by 250 meters.

This increase in resolution makes it possible observe estimated temperatures in greater detail than before. One area where this is of particular interest is the study of the urban heat island (UHI) effect (Kim, 1992). Depending on how they are designed and constructed, cities are susceptible to becoming extremely

hot, and these increased temperatures can cause and exacerbate health issues. Being able to observe the predicted air temperatures in high detail can lead to better monitoring and combatting of UHI.

The usage of LST data to estimate air temperature has been an active field of research for multiple decades. (Cresswell et al., 1999, Prihodko and Goward, 1997, Sun et al., 2005). This thesis aims to expand on the established body of work in three distinct aspects:

1. By using a type of deep learning model called a transformer as the backbone for making predictions.
2. By training the transformer on both spatially as well as temporally connected data, thereby making the transformer “spatio-temporal”.
3. By constructing the model such as to not only make a prediction for the temperature, but also to estimate the certainty it has in its prediction, thereby making it “uncertainty-integrated”.

All in all, this thesis explores new technologies and data to further develop a well established field that is becoming increasingly more relevant in light of our rapidly changing climate.

2. Preliminary materials

This section details ideas, concepts and methods that are used throughout the article (part 3).

2.1 Remote sensing data

In this thesis remote sensing refers to the acquisition of data captured via satellite or airplane, or is derived from data that was.

2.1.1 View zenith: View zenith, or viewing zenith angle, indicates the angle between the zenith of a position and the satellite that is observing it. If a satellite is observing a position directly beneath it, the angle is 0°. This angle increases as the satellite moves away from the zenith. View zenith was included as one of the ancillary data bands as the LST being measured can be affected by it (Shen et al., 2020). ECOSTRESS provides view zenith imagery as one of the bands accompanying their LST measurements. Figure 1 shows an example of view zenith data used during training and inference.



Figure 1: An example of the view zenith data for an LST image. The shifting colors indicate what the angle of that position was relative to the ISS. The values are in degrees.

2.1.2 Landcover: Landcover data informs on the makeup of the earth's surface. Pixels are classified into discrete classes, each indicative of a different type of land or environment. As mentioned in the introduction, knowledge on the type of land that an LST measurement is associated with can help with prediction of air temperature. Figure 2 shows an example of landcover data used during training and inference.



Figure 2: An example of landcover data used. Yellow areas are classified as "non vegetated", green as "vegetated" and blue as "water".

2.1.3 NDVI: NDVI stands for normalized difference vegetation index, and it is a quantification of the density and health of vegetation. NDVI is derived from remotely sensed imagery of the red and near-infrared bands. NDVI can range from -1 to 1 , with values below 0 indicating non vegetated areas, and values

above 1 representing areas of increasingly dense and healthy vegetation.

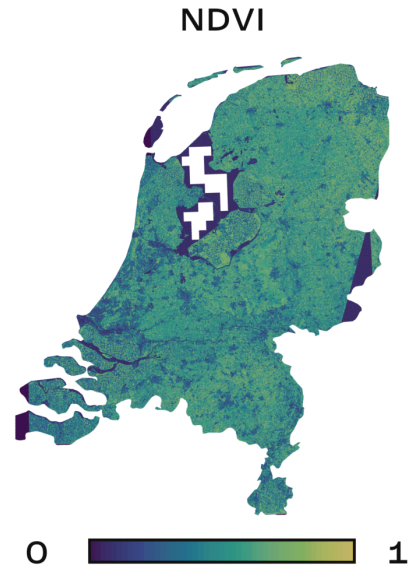


Figure 3: An example of NDVI data used. For this particular dataset values below 0 were set to 1

2.1.4 Sky view factor: Sky view factor (SVF) indicates how much of the sky can be seen from a given position. An observer in an open field will be able to see the entirety of the sky, which would correspond to a SVF value of 1 . This is in contrast to an observer standing next to a tall building. The building will obscure a significant portion of the sky, resulting in a lowered SVF value. SVF is a valuable metric as it can inform on the amount of sunlight that a position can realistically be exposed to, which is an important factor when estimating air temperature (Svensson, 2004). Figure 4 shows a sample of the SVF data used.

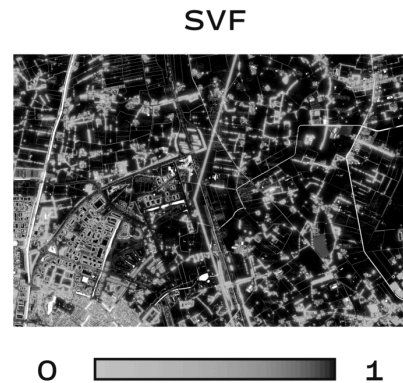


Figure 4: An example of SVF data used. A value of 0 indicates no view of the sky, while a value of 1 indicates perfect visibility.

2.2 Inference

Inference is the process of applying a trained model to new data, so as to make a prediction. In the context of this thesis it refers to the collection of data for a specific position and timestamp, and the feeding of this data into a trained model. The prediction that the model makes can then be associated with the inputs.

2.3 Embedding

Embedding is the process of representing input data as a multi-dimensional numerical vector. Data is “embedded” before being fed into deep learning layers. The goal of embedding is to ensure that data points that have similarities are closer together in multi-dimensional space, whilst the opposite is true for data points that are dissimilar. Essentially, embeddings attempt to capture the meaning and connections of data.

As an example, the word “red” might be embedded in such a way that it has similarities with the embedding for the word “turquoise”. Since both words are representative of colors, it makes sense that their embeddings share certain qualities. The embedding of “red” will also have connections to objects that are associated with that color, such as roses or fire trucks.

Determining the correct embedding for a data point is a complex problem with multiple approaches. The quality of an embedding process can determine if a model eventually succeeds in its task. The approach applied in this thesis is let the embedding itself also be done by the model. During training the model both learns how to embed the input data, and how to correctly interpret said embedded data.

2.4 Transformers

Transformers are a form of deep learning model (Vaswani et al., 2017). Introduced in 2017, they have proven capable of processing a massive variety of different forms of input data. Transformers are especially adept at making connections within and between data sources.

2.4.1 Self Attention: The component in transformers responsible for their abilities to make connections is the attention mechanism. Attention allows a model to determine the importance of an element in a sequence with respect to all other elements. As an example, in the sentence: “The red ball flew through the air”, the word “red” will have a greater association with the word “ball” than with the other words.

Attention is often applied many times in a row. This repeated analysis makes it possible to capture the subtle connections latent within the input data.

2.4.2 Cross attention: In self attention, the components of an input are compared with each other. In contrast cross attention compares the parts of one sequence with the parts of a different one. This can allow for finding connections between different types of data. A classic example is translation of languages, where cross attention will associate a word with its translation.

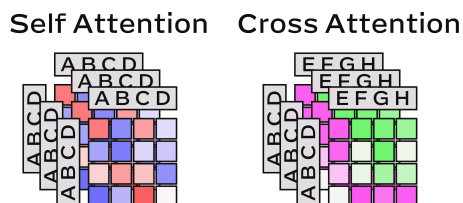


Figure 5: A simplified representation of self attention and cross attention. The repeated images signify the layering of the attention blocks.

2.5 SWIN transformer

Shifted WINDOW (SWIN) transformers are a relatively recent breed of transformer that specializes in the analysis and processing of image data (Liu et al., 2021). Images are first divided into small patches, after which attention is applied within a window, which is a grouping of patches. Next the windows are “shifted”, as can be seen in Figure 6. After shifting attention is again applied within the newly formed windows. This shifting causes the patches to be exposed to a different selection of their peers.

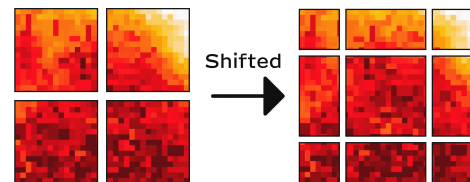


Figure 6: An image representing how the windows of attention are “shifted”.

A second aspect of SWIN transformer is the merging of patches. After attention has been applied amongst patches, they are merged together to form larger patches. This process repeats, with each iteration processing progressively fewer total patches of increasing size. This is illustrated in Figure 7. By applying attention at both small and large scales within an image, SWIN transformers can process both the fine details and large features within an image.

SWIN



Figure 7: A simplified representation of how a SWIN transformer processes images at changing patch density.

3. Article

Page left intentionally Blank

Air temperature estimation through thermal satellite imagery using uncertainty-integrated transformers

Lars van Blokland¹

¹ Delft University of Technology – Delft, the Netherlands – L.W.vanBlokland@student.tudelft.nl

Keywords: Remote sensing, Deep learning, Spatio-temporal transformer, Uncertainty integrated, LST to AT.

Abstract

Accurate estimates for air temperatures in urban environments can help with timely and precise action against the urban heat island (UHI) effect. Uncertainty-aware spatio-temporal transformers are an ideal candidate model for producing such predictions. This study details the implementation and testing of a transformer model that combines remotely sensed land surface temperature (LST) data with in-situ sensor readings to predict air temperature values for the entirety of the Netherlands. The three best models out of the 21 trained attained an averaged test-set error of 1.365° Celsius MAE. Model inference has produced Geotiffs of air temperature and uncertainty predictions for all of the Netherlands, at a 70m by 70m pixel resolution. NASA's ECOSTRESS dataset supplied LST imagery and assorted ancillary bands, ESA's Copernicus provided NDVI and Landcover data, and sensor readings were acquired from the royal Dutch weather service (KNMI). Overall this study details the design for a spatio-temporal transformer model that produces uncertainty-aware air temperature estimations.

1. Introduction

Accurate monitoring of air temperature in urban environments is critical to understanding and combatting the urban heat island effect (Deilami et al., 2018, Kim, 1992). Traditional methods of interpolating (Bigg, 1991) with in-situ sensor data may not be able to provide the coverage and resolution necessary for good analysis (Weng et al., 2014). Using remotely-sensed land surface temperature (LST) measurements to predict air temperature can allow for a much finer coverage when compared to interpolation of in-situ sensor data (Benali et al., 2012). The NASA ECOSTRESS (NASA, 2018) mission captures LST data at a higher spatial resolution than predecessors such as MODIS (Salomonson et al., 1989), allowing for more accurate air temperature estimates.

Air and surface temperature are correlated (Kawashima et al., 2000), but their relation is affected by numerous factors, including the amount of vegetation, exposure to sunlight, wind speed and others (Svensson, 2004, Yan et al., 2014). Additionally this relation changes over time, affected by both daily and yearly cycles (Naserikia et al., 2023). The availability and quality of remotely sensed data can also vary wildly (Yoo et al., 2018), often requiring repair or post processing. Depending on the orbit that a remote sensing instrument follows, subsequent captures of a position may be days apart. Clouds and other atmospheric interference can also cause data to be unusable or less certain. The result of these relations and limitations mean that a predictive model needs to not only incorporate a wide range of different data types, but also be resilient to temporally sparse and potentially unreliable inputs. Additionally, it is beneficial if a model incorporating these data can also report on the confidence it has in a particular prediction. This uncertainty value provides valuable context, helping users make informed decisions when utilizing model predictions.

The ability to understand latent relations across a range of data modalities and data qualities, is something that transformer models (Vaswani et al., 2017) have proven to be capable of. Transformers make use of self-attention to find connections between inputs without the need to split them up. This ability

makes transformers well suited for the task of combining multi-source remote sensing and in-situ sensor data. Many recent studies working with transformers in remote sensing focus on classification (Bazi et al., 2021, He et al., 2022, Roy et al., 2023) and change detection (Chen et al., 2021, Jiang et al., 2023, Zhang et al., 2022), wherein the transformer is required to fuse data of different modalities together. Change detection in particular makes use of spatio-temporal transformers, that apply attention across both spatial and temporal axes.

Although usage of transformers has become ubiquitous within remote sensing, prediction of air temperatures remains an under represented application. However, given that ground-air temperature dynamics are complex processes spread across both time and space, spatio-temporal transformers make for excellent model candidates. This study has employed an uncertainty-aware spatio-temporal-transformer described in (Rafiee, 2026, under review) as backbone, and has experimented with a different approach towards cross-attention masking. The model will utilize transformers to combine multiple remotely-sensed bands together with in-situ sensor data. Experiments will focus on determining how model design choices and data type inclusions affect the performance of the model in different spatial and temporal contexts. Of special interest is gauging the models quality and certainty of its predictions across a variety of urban environments.

2. Study area and datasets

2.1 Study area and data window

The study area comprises the entirety of the Netherlands. All data used was captured between December 2021 and December 2025. This range ensures that all seasons are present multiple times and allows for the representation of multi-year weather cycles.

2.2 In situ sensor data

Hourly air temperature, wind speed and wind direction measurements used to train the model were acquired from 269 stations spread across the Netherlands. 35 of these stations are

maintained by the KNMI (KNMI, 2026), the Dutch weather service. The remaining 234 stations are run by volunteers through the KNMI WOW (WOW, 2026) program. Stations were selected to ensure a good distribution of differently urbanized areas. Stations too close to large bodies of water were omitted.

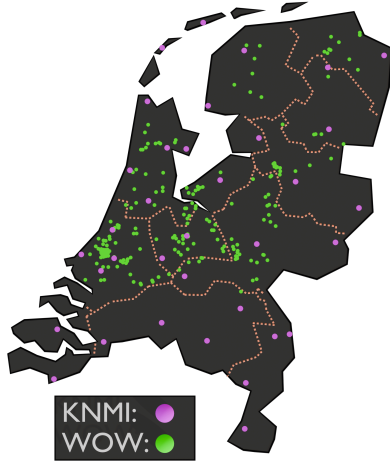


Figure 1: The position of all 269 stations

2.3 Remotely sensed data

2.3.1 LST: LST image data was acquired through NASA’s ECOSTRESS program (NASA, 2018). ECOSTRESS data has a resolution of 70 by 70 meters per pixel. Accuracy of these measurements is recorded in supplemental products, which were integrated into the model training process. ECOSTRESS data is collected by an instrument attached to the International Space Station (ISS). The orbit of the ISS results in different data capturing behavior depending on the latitude of the region being observed. Although the time between captures for the Netherlands is short, the amount, quality and subject of data collected varies greatly. The average time between a high quality capture for the 269 station positions was 6.5 days, with values ranging between 1 hour and 31 days.

Additional bands provided by the ECOSTRESS product that were utilized during the training process include:

- **QC** (quality control), which informs on the quality of individual pixels. QC datasets consist of 5 bands each
- **Height**, indicating the elevation in meters
- **EmisWB**, the broadband emissivity associated with the LST estimate
- **LST_err**, the uncertainty of the LST estimate
- **View_zenith**, the view zenith of each LST pixel

Since they are provided together, the availability and resolution of these ancillary bands is identical to that of the LST band. Information on these ancillary bands can be found via (Cawse-Nicholson et al., 2023). A more detailed description of the QC band is available in (Hulley and Freepartner, 2019).

2.3.2 Normalized Difference Vegetation Index (NDVI): NDVI data was downloaded via ESA’s Copernicus browser, and was derived from bands captured via ESA’s Sentinel 3 satellite. Download resolution was set to 70 by 70 meters per pixel to match ECOSTRESS data. Since NDVI changes less frequently than LST, NDVI data was only acquired for a subset of the timestamps where LST data was available (Copernicus, 2020a).

2.3.3 Land cover: Also downloaded via the Copernicus browser was land cover data, derived from data captured via

ESA’s Sentinel 2 satellite. The data was obtained in the same manner as NDVI and at the same resolution. Since land cover also changes infrequently, it too was only downloaded for a subset of the time window (Copernicus, 2020b).

2.3.4 Sky view factor: Sky view factor (SVF) data was acquired via the Dutch government’s data api (KNMI, 2026). This SVF dataset has a resolution of 0.5 meters per pixel, but was not used as an image by the model. Instead SVF datasets were sampled at station positions, and the found values were incorporated as feature data.

2.3.5 Wind speed and direction: During model inference station data is unavailable, necessitating the need for an alternative. The ERA5 (Hersbach et al., 2023) dataset provides, amongst others, bands for the wind strength in west-east and north-south directions. These bands were sampled during inference to provide wind data.

2.4 Data preparation

2.4.1 Data pre-processing: Image datasets were acquired as tiles, and these were merged based on temporal proximity. Merged datasets were then sampled at station positions to create smaller images directly surrounding the target. The resolution of these samples was 32 by 32 pixels, and samples with less than 80% valid pixels were discarded. This sampling process was done for all image types except SVF. For both image and feature data, values were normalized to be between 0 and 1, with invalid values being set to -1. Merging and image sampling was done using Rasterio (Gillies and others, 2013).

The 32 by 32 pixel sample size was chosen to balance processing speed and amount of considered data. 64 by 64 and 128 by 128 were also tested, but these sizes did not increase model performance. Normalization of values was done to facilitate stable model training.

Figure 2 and Figure 3 show the distribution of data before and after sampling. The first set of bars in each of the figures indicates how many LST datasets belonged to a particular hour or month. The distribution of both hours and months adhere to a distinct pattern. Hours start lower, then increase throughout the day, peaking at noon, and then diminishing again when approaching midnight. This suggests that producing ECOSTRESS tiles is easier during the day, at least for the Netherlands. Months alternate between those with many datasets and those with few. One possible explanation is that the orbit of the ISS causes periods wherein regions are either over or under-sampled.

The second set of bars in Figure 3 and Figure 2 indicate how many samples were taken from each period. If every dataset produced an equal number of image samples, the first and second columns would be identical. Since this is not the case it can be assumed that particular moments are more likely to generate valid samples than others. Noteworthy is that the sampling process seems to accentuate patterns present within the dataset distribution. This could signify that moments with more data also have inherently higher quality data as well.

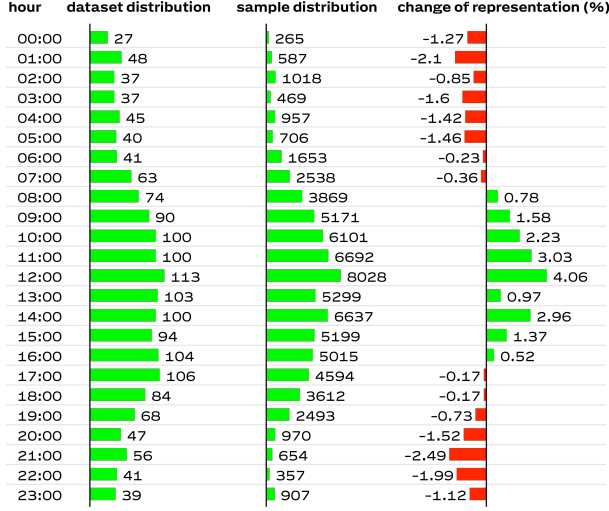


Figure 2: The distribution of data, before and after sampling, organized by hour.

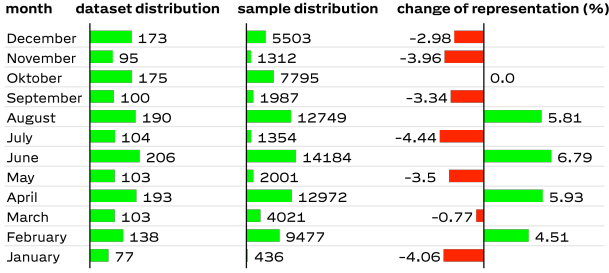


Figure 3: The distribution of data, before and after sampling, organized by month.

2.4.2 Data instance assembly: Data instances serve as the models input. A data instance is comprised of three components: the input images, the input features and the ground truth air temperature value associated with the input data. Data instances are built up around the LST samples extracted from the datasets. Each of these samples is associated with a timestamp and a (station) position. These two values are used to find or calculate all other data needed to complete the instance. This searching step is necessary, since data not from the ECOSTRESS captures is likely to have a differing temporal resolution. The process is visualized in Figure 4.

Once all data instances have been made, stacks can be created. Each data instance acts as the foundation for its own stack. For each data instance, the N instances that precede it temporally and share a position are found and appended to the stack. For elements in a stack other than the first, the time delta to the first element is calculated and added to the data instance. This stacking adds a temporal element to the data.

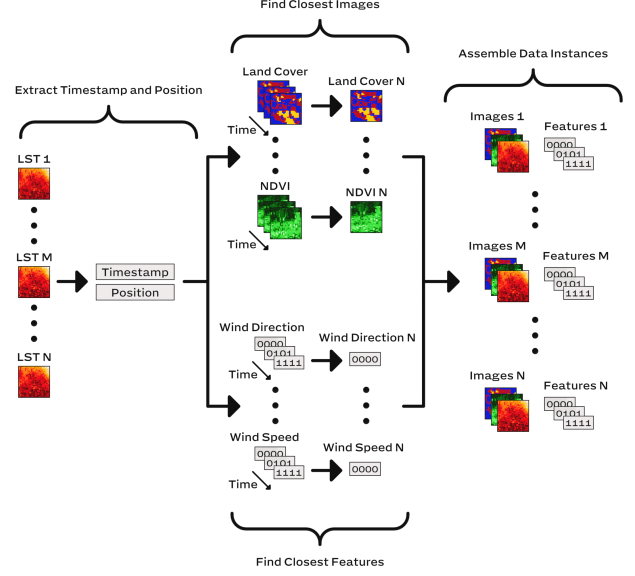


Figure 4: The process for assembling the data instances

3. Methodology

3.1 Model Architecture

The model is comprised of two main components, which process images and features, and one ancillary component, which processes mask data. Each component is fed its own input data, embeds this data and feeds these embeddings into a transformer layer. The three outputs of these transformer layers are combined via cross-attention and run through a final linear layer, which outputs the model predictions. An overview of the model architecture can be seen in Figure 5. The model was created using the PyTorch library (Paszke et al., 2017).

3.1.1 Image learning: Images are processed in stacks. Each stack is comprised of N data instances, with each data instance containing M images. Data instances always start with an LST image. An example for $N = 2$ and $M = 3$ would be:

$$[[\text{LST}, \text{QC}, \text{NDVI}], [\text{LST}, \text{QC}, \text{NDVI}]] \quad (1)$$

The values for N and M depend on the model setup conditions. Each data instance is embedded separately via a convolutional layer. This embedding process splits the images into distinct patches, and then embeds patches together that represent the same area in the image. In parallel to this, the temporal information belonging to each data instance is also embedded. The temporal embeddings for all patches in the same data instance are identical, as these patches share a timestamp. The embedded instances are then combined with their respective temporal embeddings after which they are concatenated together at the patch level. This process is illustrated in Figure 6.

These concatenated patches form the input to the SWIN transformer (Liu et al., 2021). The transformer applies attention between the created patches, thereby learning spatial and temporal connections latent within the embedded data. After patches have been exposed to each other via attention, they are merged, creating a smaller number of patches of a higher dimension. This process repeats a set number of times after which the result is flattened into a high-dimensional vector.

3.1.2 Feature learning: Features are structured the same way as images: each feature stack contains N instances, and each instance contains W features. While the number of instances in a stack will always be identical for images and features, the number of elements in an instance does not have to be.

Feature instances are first concatenated together before being embedded by a linear layer. This embedding is then fed into a multi-head self-attention transformer, whereby the embedding attends to itself multiple times. The final output is a high-dimensional vector.

3.1.3 Temporal information: Temporal information was encoded in both the image and feature data. For the features this information was simply concatenated together with the other feature data such as wind speed and SVF. For the images the temporal information is first embedded into the patch embedding dimension, and then fused with the image patch embeddings. Hour-of-day and day-of-year were passed through the sine and cosine functions so as to encode the information in a cyclical manner. Doing so better emphasizes the behavior of temperature across days and years:

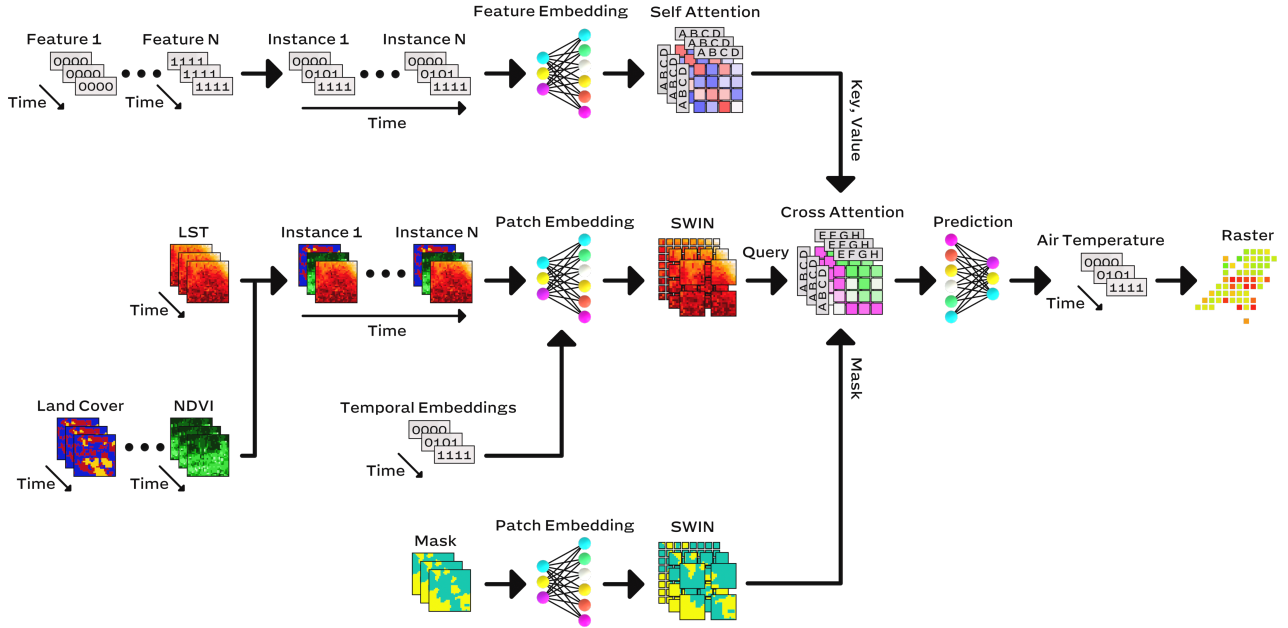


Figure 5: The model architecture.

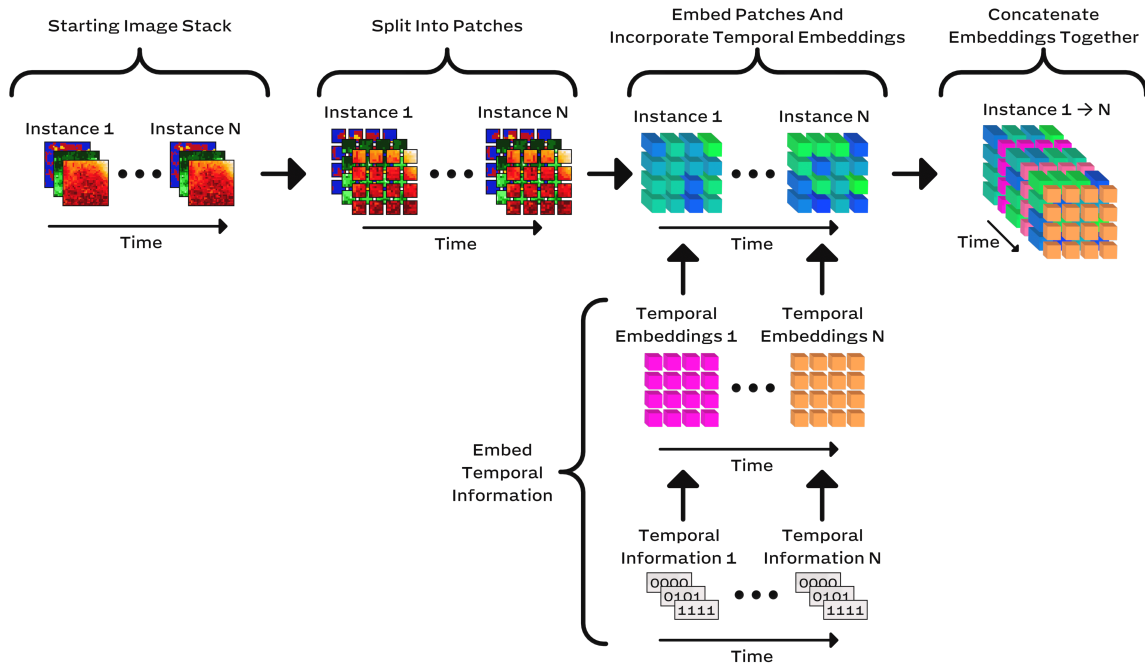


Figure 6: The image embedding process.

$$\sin\left(2\pi\frac{\text{HoD}}{24}\right), \cos\left(2\pi\frac{\text{HoD}}{24}\right) \quad (2)$$

$$\sin\left(2\pi\frac{\text{DoY}}{365.5}\right), \cos\left(2\pi\frac{\text{DoY}}{365.5}\right) \quad (3)$$

Where:

- **HoD** indicates the hour that a capture occurred
- **DoY** indicates the day that a capture occurred

These four values were accompanied by a delta value indicating the temporal distance between subsequent captures:

$$\log(1 + \triangle t) \quad (4)$$

Where:

- $\triangle t$ is the time difference in hours

The process for fusing the patch and temporal embeddings is as follows:

$$\mathbf{e}_{\text{fused}} = \mathbf{e}_{\text{patch}} \odot \gamma + \beta \quad (5)$$

Where:

- $\mathbf{e}_{\text{patch}}$ are the embedded patches
- γ and β are the outputs of embedding the temporal information. Both have dimensions equal to $\mathbf{e}_{\text{patch}}$

3.1.4 Mask learning: Unlike the standard image inputs, mask data is not stacked. Only the first data instance in a stack will have accompanying mask data. Since the LST image of the first data instance in a stack carries the highest weight, it was deemed unnecessary to provide mask data for LST images of later data instances.

Since all mask images are of the same timestamp, they are not temporally embedded. After patch embedding they are directly fed into a SWIN network, the output of which is a vector with dimensions equal to that of the image network.

3.1.5 Cross attention: In self-attention the query, key and value matrices are derived from a singular input. Cross attention contrasts this by utilizing two data sources. These two data sources can be of different modalities, making cross attention useful for fusing data of differing types. The formula for cross attention is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (6)$$

Where:

- Q is the query
- K is the key
- V is the value
- d_k is the dimension of the key

In the model, cross attention is performed with the outputs of the image, feature and mask components. Image outputs are used as queries, feature outputs as keys and values, and finally uncertainty data is used as a mask. The output of the cross-attention layers is a vector with dimensions equal to those of the query input.

3.1.6 Prediction: Predictions are made by feeding the output of the cross-attention network into a series of linear layers. These linear layers reduce the dimensionality of the data until it has reached the desired size. The model predicts the air temperature for the position and timestamp specified by the LST image

of the first data instance in a stack. The model also predicts temperature values for the next 4 hours directly following the target timestamp, giving an estimate for how the air temperature will change. Each temperature prediction is also accompanied by a prediction for the variance of the temperature. This results in a total of 10 predictions per data stack.

Weights were assigned to the losses of hourly predictions to encourage the model to prioritize the accuracy of earlier hours over later ones.

3.2 Model performance

Models were trained for 100 epochs on ~51,000 data instances. Of these instances 70% were used for training, 20% for validation and 10% for testing. All models used the same data split. During the training process, models were saved if they performed better on the validation set than any of their previous iterations. After training for 100 epochs, models predicted for the test set once using the last saved model, and the performance here was used to rank and evaluate the models in Section 5. Batch size was set to 256, and the number of dataloader workers was set to 16.

3.2.1 Loss function: The loss function used during the training of the model is Gaussian negative log likelihood (gnll) (Nix and Weigend, 1994):

$$\text{loss} = \frac{1}{2} \left(\log(\sigma^2) + \frac{(\text{input} - \text{target})^2}{\sigma^2} \right) \quad (7)$$

Where

- **input** is the predicted air temperature
- **target** is the ground truth air temperature
- σ^2 is the predicted variance of the predicted air temperature

gnll treats the target values as samples from a Gaussian distribution, with the expectation and variance being predicted by the model. The model will learn that in cases where it is less certain of accurately predicting the target air temperature, it can increase the variance to lower the resulting loss. However raising the variance whilst its predictions are accurate is detrimental. Thus the model learns to raise the variance only when it is certain that its air temperature estimation is untrustworthy.

Predicting a variance provides additional context for the air temperature estimations, mainly giving a measure for the confidence that the model has in a particular prediction. In the context of this research, it enables identifying which regions are harder to accurately predict air temperatures for, as an example.

3.3 Inference

The timestamp **2022-08-13 12:25:24** was chosen for inference, as the LST image at this moment, and those of the ones preceding it are of very high quality. Additionally the images were captured within a very short timeframe, with the time deltas to the primary LST image being 2 and 23 hours for the secondary and tertiary images respectively. These LST images are visible in Figure 7. One potential downside of processing in this particular moment is that it was classified as a heatwave, with extremely high temperatures and almost no cloud cover (KNMI, 2022). Predicting for an outlier situation such as this makes it less certain how reliable the predictions are.

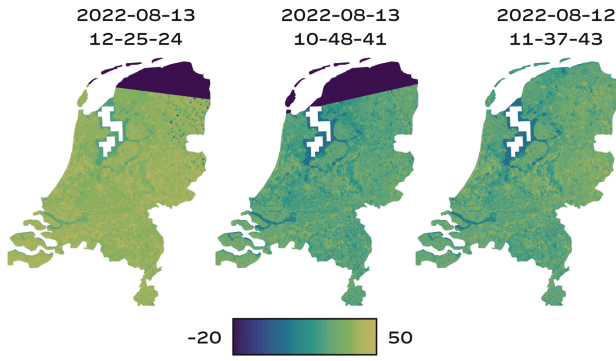


Figure 7: LST images for the 3 timestamps used during inference. Values are in Celsius.

Points were generated across the surface of the Netherlands, spaced 70 meters apart. During inference, these points were grouped together into batches of 10000. For each point in a batch, a multiband Geotiff containing all bands for all three timestamps was sampled at the point position. These stacked samples were then filtered based on which bands were needed by the model performing inference, and then rearranged into the order expected by the model. In parallel, databases containing pre-sampled SVF and wind data were queried, and the acquired data was combined with the image samples to form the model inputs. After the model finished predicting values for all of the points, these predictions were written to a multiband Geotiff.

4. Experiments

In total 21 model setups were tested. One of these setups was the “base” model, from which all others were derived. These derived models would change a single model parameter of the base model, so that the impact of said parameter could be gauged. The gnll losses for these models can be seen in Figure 8 and the processing times in Figure 11.

The base model was trained with the following parameters:

- LST and all ancillary image types (NDVI, height, emisWB, view_zenith, LST_err, QC, landcover) were used as image inputs.
- The image inputs were combined with temporal embeddings.
- All feature types (wind direction, wind speed, temporal information, sky view factor) were used as feature inputs.
- A mask was used during cross attention, and it was derived from the QC and LST_err image types.
- Each input stack comprised two data instances, one for the current timestamp and another for the timestamp preceding it.
- The losses of the 5 predictions had the following weights: [0.33, 0.27, 0.2, 0.13, 0.07], in order of current timestamp to +4 hours.
- Uses [3, 6, 12, 24] heads of attention for SWIN, has a SWIN embedding dimension of 48 and has 200 SWIN output classes.

4.1 Ablation testing

A majority of the setups tested withheld one or more of the inputs from the model to ascertain the importance of that data type. The following models were of this kind:

Images:

- **no_landcover**
- **no_NDVI**
- **no_emisWB**

- **no_height**
- **no_view_zenith**
- **no_QC**
- **no_LST_err**
- **no_temporal_embedding**
- **no_ancillary**: Only LST was used as a model image input, however QC and LST_err were still used to create a mask.
- **no_ecostress_ancillary**: The ECOSTRESS ancillary bands are: emisWB, LST_err, view_zenith, QC and height.
- **no_ndvi_no_landcover**

Features:

- **no_wind**: Both wind speed and wind direction were omitted.
- **no_svf**
- **no_feature_time**: The time features are mentioned in Section 3.1.3, all five were omitted.

Mixed:

- **no_ancillary_images_no_svf**

4.2 Temporal stack testing

The base model had a temporal stack size of 1, meaning that each stack contained a data instance for the current timestamp, and the first available one before it. Two models were tested that changed this parameter:

- **temporal_stack_0**: Each stack only contained a single data instance, the one for the current timestamp.
- **temporal_stack_2**: Each stack contained 3 data instances, one for the current timestamp and then the two that directly preceded it.

4.3 Mask testing

To promote the incorporation of data quality data in the cross attention process, specific bands are processed into a mask that is used to manipulate the attention between image and feature data. In the base model only the QC and LST_err bands are used to produce this mask. The following models experimented with different setups for the mask.

- **no_mask**: The model was trained without a mask during cross attention.
- **all_image_as_mask**: All ancillary image types were used to create the cross attention masks.
- **all_ancillary_image_as_mask_not_image**: All ancillary image types were used to create the cross attention masks, however they were not used as image inputs.

4.4 Training setup

Training was conducted on a machine with the following specifications:

- NVIDIA Corporation GA102 [GeForce RTX 3090]
- Ryzen ThreadRipper CPU

5. Training Results

The following section details the results of model training. All listed losses were achieved on the test set.

5.1 Overall model performance

Losses for the tested models can be seen in Figure 8. Immediately evident is that nearly all tested models outperformed the base model. Removal of each individual image input lead to an improvement, which was also true for the feature inputs with exception of the time feature. The contrast between the influence of removing the time feature and the temporal image embedding, despite being essentially the same information, in-

indicates that the temporal embeddings are not succeeding in their task of informing the model when predictions are being made.

The performance of the two temporal stack models shows that having information on previous LST captures is beneficial, but only up to a point. The average time between data instances was:

- 143 hours between the first and second data instance.
- 285 hours between the first and third data instance.

The increased performance of the base model over the temporal stack 0 model indicates that, despite an average gap of 143 hours between data instances, the model is still able to glean useful information. This is in contrast to the temporal stack 2 model. Either 285 hours is too large a gap for information to still be relevant, or the 50% increase in information (compared to the base model) that needs to be processed by the same amount of model parameters is inhibiting proper learning of said information.

This overcrowding may also be an explanation for the increased performance when features are omitted. Removal of an image type whose contribution to accuracy is limited may allow the model to better focus on more relevant information. The order of the individual image ablation models could give an indication of how relevant each image type is. This would suggest the NDVI contributes the most to accuracy, whilst view_zenith contributes the least.

The performance of the mask model shows that the QC and LST_err images types are not helping when integrated as a mask. However, the accuracy of the “all_image_as_mask” model indicates that the mask can contribute significantly if it is comprised of the correct information. One possibility is that the mask is alleviating the potential overcrowding of the model parameters. The mask layers effectively increase the amount of parameters that the model can use to learn the task. The “all_ancillary_as_mask_not_image” model which is effectively the inverse of the “no_mask” model, have almost identical losses. This indicates that the manner in which data is supplied (either as an input to cross attention or as a mask) is less relevant than the volume of information. The performance of the “all_image_as_mask” clearly shows that simply giving the model more room to process the same data leads to improvements.

The good performance of models such as “no_ndvi_no_landcover” and “no_ancillary_images_no_svf” indicates that it is not necessary to have a full set of ancillary bands to create an acceptable model. This is a desirable quality as it shows that regions without good remotely sensed data can get by using only ECOSTRESS data, which is available for free and has (near) global coverage.

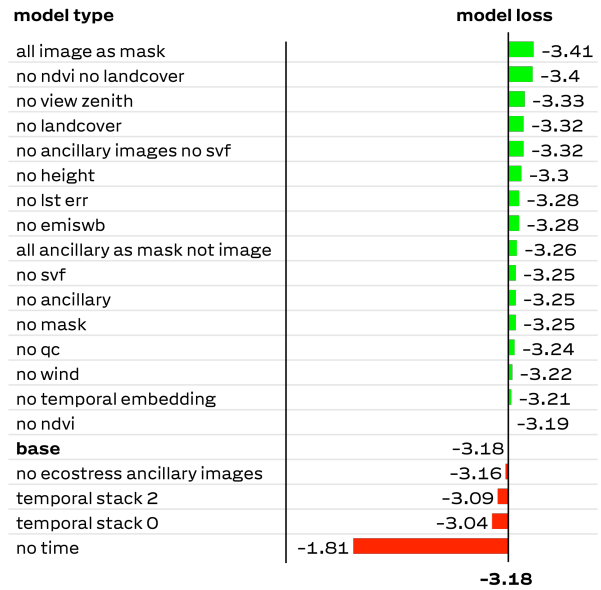


Figure 8: The performance of the 21 tested models. The colored bars indicate the distance to the loss of the base model

5.2 Seasonal model performance

Figure 9 and Figure 10 show the averaged performance of all models across the period of a full year and a single day respectively. Note how the samples follow the distributions of the overall dataset as described in Section 2.4.1. There do not seem to be any significant patterns within the measured performances. In both figures a number of instances deviate from the mean significantly, but these are generally confined to periods with very low sample counts. The exception to this is the month of October, which has both a high sample count, and is at least one standard deviation away from the mean.

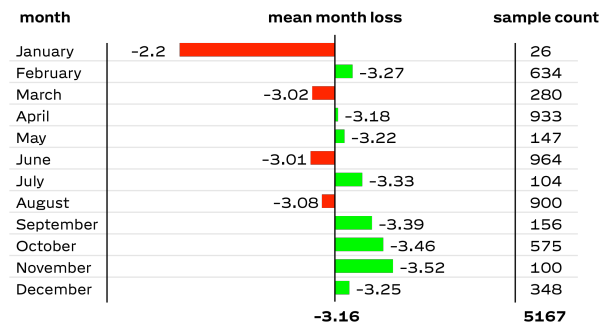


Figure 9: Averaged performance of all models across a year. The colored bars indicate the distance to the mean loss.

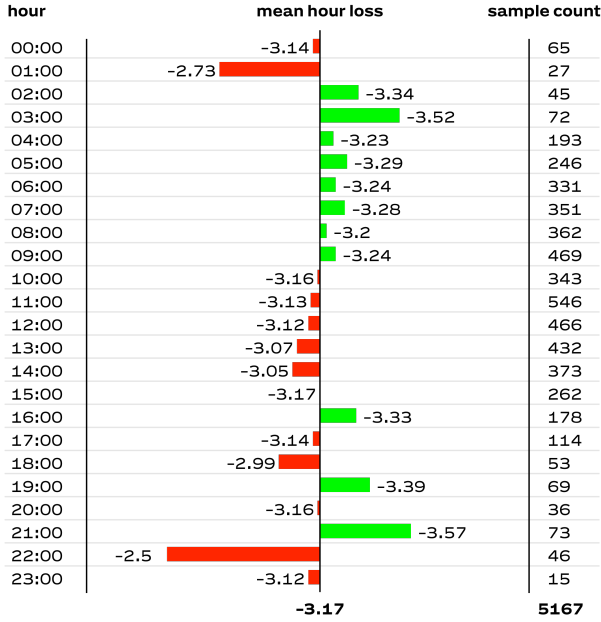


Figure 10: Averaged performance of all models across a day. The colored bars indicate the distance to the mean loss.

	standard deviation
month	0.222
hour	0.146

Table 1: The standard deviations for Figure 9 and Figure 10.

5.3 Model training speeds

Figure 11 shows how much time each model needed to complete a single epoch. Models with reduced image inputs processed faster than those with more. This is not entirely surprising, as the computational complexity of the SWIN components is greater than any of the other layer types used in the model. The impact of removing feature data is almost non-existent, with the “no wind”, “no svf” and “no time” models having times nearly identical to that of the base model. The effect of changing the size of the data instance stack is clearly visible by comparing the “temporal stack 0”, “base”, and “temporal stack 2” processing speeds, increasing in steps of approximately 27 seconds for each added layer in the stack.

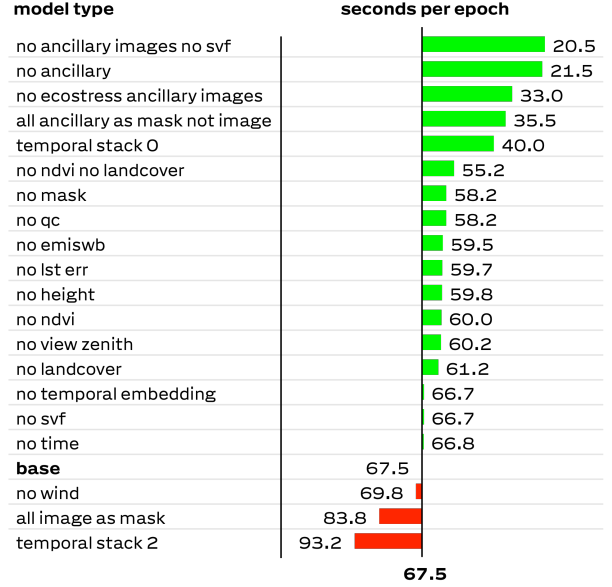


Figure 11: The amount of time in seconds necessary for every model type to complete a single epoch. The colored bars indicate the distance to the base model. The listed values combine the training and validation times.

5.4 Model training behavior

Figure 14 shows how the validation loss of the models behaved during training. Most of the models achieved their best validation score around the 50th epoch. After this the model would start to overfit on the training set, as indicated by losses that generally get worse after their best performance. Notable is that the two best performing models “no_ndvi_no_landcover” and “all_image_as_mask” both experienced a sudden improvement near the 50th epoch.

5.5 Increased model size experiments

Additional models were trained and tested specifically to ascertain if increasing the amount of model parameters would improve performance. For these experiments variations of both the “base” and “all_image_as_mask” models were made. The tested models were:

Variations on the “base” model:

- **base_swin_big**: Uses [4, 8, 16, 32] attention heads during SWIN, instead of [3, 6, 12, 24].
- **base_96**: Uses a SWIN embedding size of 96, instead of 48.

Variations on the “all_image_as_mask” model:

- **all_image_as_mask_400**: Uses 400 output classes for the SWIN components instead of 200.
- **all_image_as_mask_cross_heads_6**: Uses 6 heads of attention during cross attention instead of 3.
- **all_image_as_mask_swin_big**: See above.
- **all_image_as_mask_96**: See above.
- **all_image_as_mask_96_300**: Uses 300 output classes and an embedding size of 96.

The test performances for these models can be seen in Figure 12. Both of the “base” model varieties improved against their predecessor, suggesting that the increased capacity is beneficial. This was not the case for the “all_image_as_mask” model which did not gain any improvements from altering the model setups. One possibility for this is that the increase in model parameters only allowed the model to overfit onto the data more than the

standard version. The results indicate that managing the correct number of model parameters is a delicate balance between poor performance and overfitting.

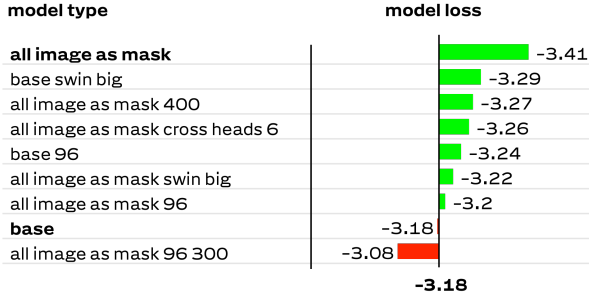


Figure 12: Performance of the additional models tested compared to the “base” and “all_image_as_mask” models

Figure 13 shows the result of an additional data type ablation test on the “all_image_as_mask” model. The results indicate that all data types can contribute towards performance if the model has enough space.

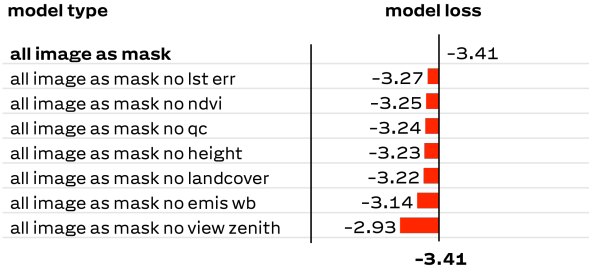


Figure 13: Performance of variations of the “all_image_as_mask” model when data types are omitted.

6. Inference Results

Inference was executed on the three best performing models: “no_ndvi_no_landcover”, “all_image_as_mask” and “no_view_zenith”. The inference process results in a Geotiff file with ten bands total. The first five bands contain temperature predictions for the hours 1-5, and the last five bands contain variance predictions for the hours 1-5. Figure 15 and Figure 16 visualize the predicted values for the first and fifth hours respectively. Visible is how the temperatures decrease across the five hour period while the variance rises. Hour 1 occurs at **12:25:24** and hour 5 at **16:25:24**, making the decrease in temperature desired, as it properly models how temperatures drop as the sun moves away from its apex. The lowering of the variance indicates that the model becomes less confident in its predictions as it forecasts progressively further into the future.

A concerning feature noticeable in Figure 15 and Figure 16 is that the temperature and variance maps have a visible degree of similarity. This indicates that the variance is somehow correlated to the temperature. Although both values are predicted using the same underlying data, the loss function described in formula (7) eliminates the absolute value of predicted temperatures by subtracting it from the ground truth. This ensures that the predicted temperature and predicted variance never interact directly. One possibility for this correlation could be the circumstances of the particular moment inference was performed for. As described in Section 3.3, the inference timestamp occurred during a heatwave, thereby exposing the model to data it may not be overly familiar with. When predicting temperatures that

deviate significantly from the mean, the model is encouraged to raise the variance, as doing so will lower the loss it incurs in the event that its prediction missed the mark. This risk averse behavior could lead to temperature and variance being correlated.

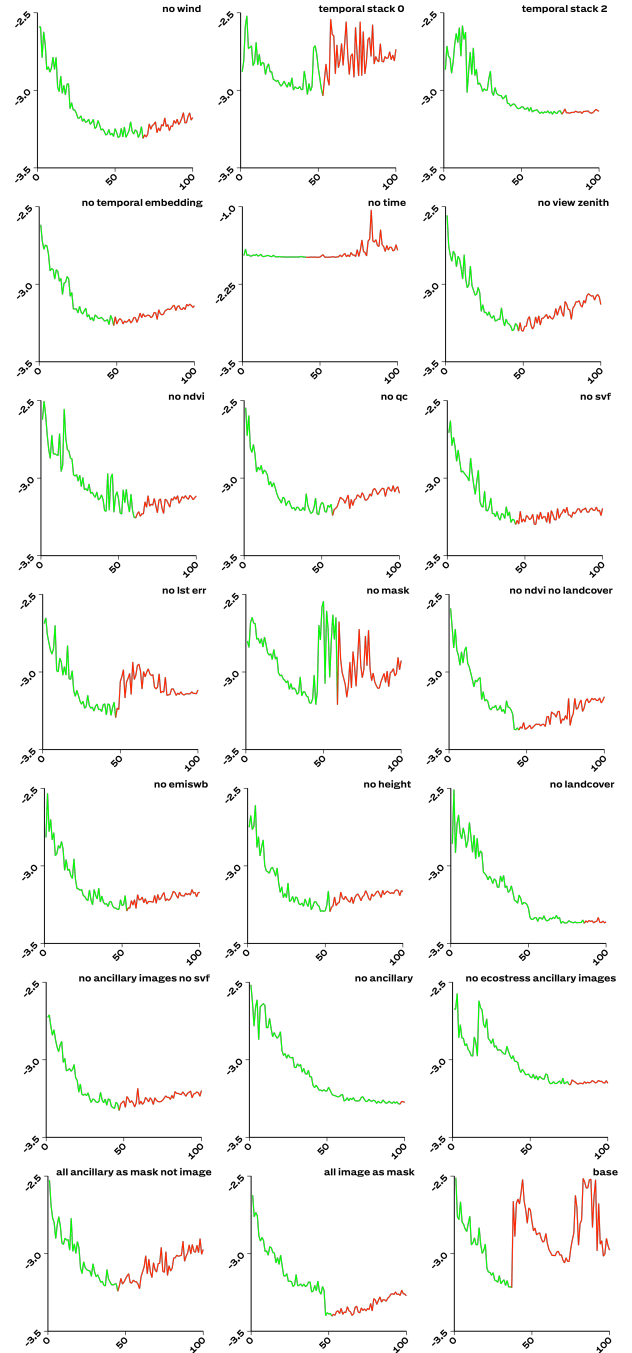


Figure 14: The behavior of the model validation losses during training. The Y axis contains the validation loss, whilst the X axis lists the epoch number. The lines changing color indicate at which point during training the best validation loss was achieved. Note that due to the exceptionally poor performance of the “no time” model, its loss is scaled between -1 and -3.5, instead of -2.5 and -3.5 for all the other models.

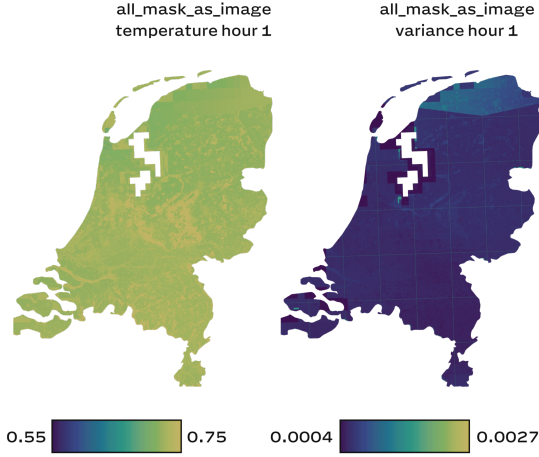


Figure 15: The predicted temperature and variance maps for hour 1 of the “all_image_as_mask” model. Temperature values are a factor between 0 and 1, denoting a temperature between -20 and 50 degrees Celsius.

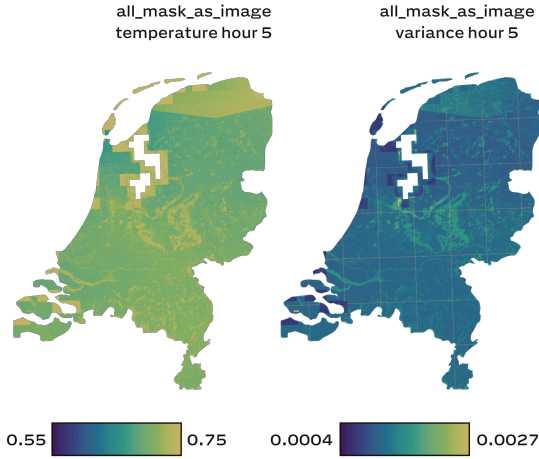


Figure 16: The predicted temperature and variance maps for hour 5 of the “all_image_as_mask” model. Temperature values are a factor between 0 and 1, denoting a temperature between -20 and 50 degrees Celsius.

Figure 19 shows, for the three best models, the temperature, variance and difference between the two. This difference was calculated with the following formula:

$$\text{difference} = \text{abs}(\text{temp}_n - \text{abs}(\text{var}_n - 1)) \quad (8)$$

Where

$$\text{temp}_n = \frac{\text{temp} - \min(\text{temp})}{\max(\text{temp}) - \min(\text{temp})} \quad (9)$$

$$\text{var}_n = \frac{\text{var} - \min(\text{var})}{\max(\text{var}) - \min(\text{var})} \quad (10)$$

After normalizing the temperature and variance maps between their minimum and maximum values, the variance is subtracted from the temperature. The resulting map highlights areas where temperature and variance are different in scale. If a model produces temperatures and variances that correlate with each other, the difference map will be devoid of features. The differ-

ence map for the “no_ndvi_no_landcover” model in Figure 19 is almost entirely flat, signifying a heavy correlation between the predicted temperatures and variances. Although the two other models have more visible differences, they are still far from ideal.

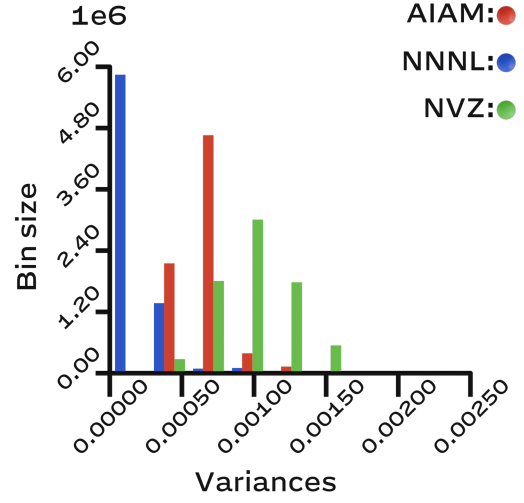


Figure 17: Histogram of the inference variance values of the three best performing models. Model names were abbreviated: NNNL: “no_ndvi_no_landcover”, AIAM: “all_images_as_mask”, NVZ: “no_view_zenith”

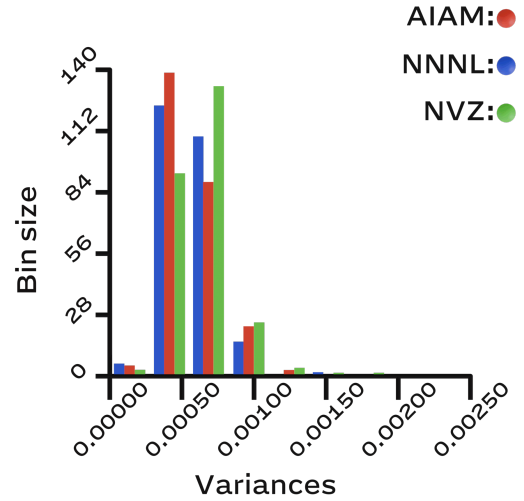


Figure 18: Histogram of the training variance values of the three best performing models. Model names were abbreviated: NNNL: “no_ndvi_no_landcover”, AIAM: “all_images_as_mask”, NVZ: “no_view_zenith”

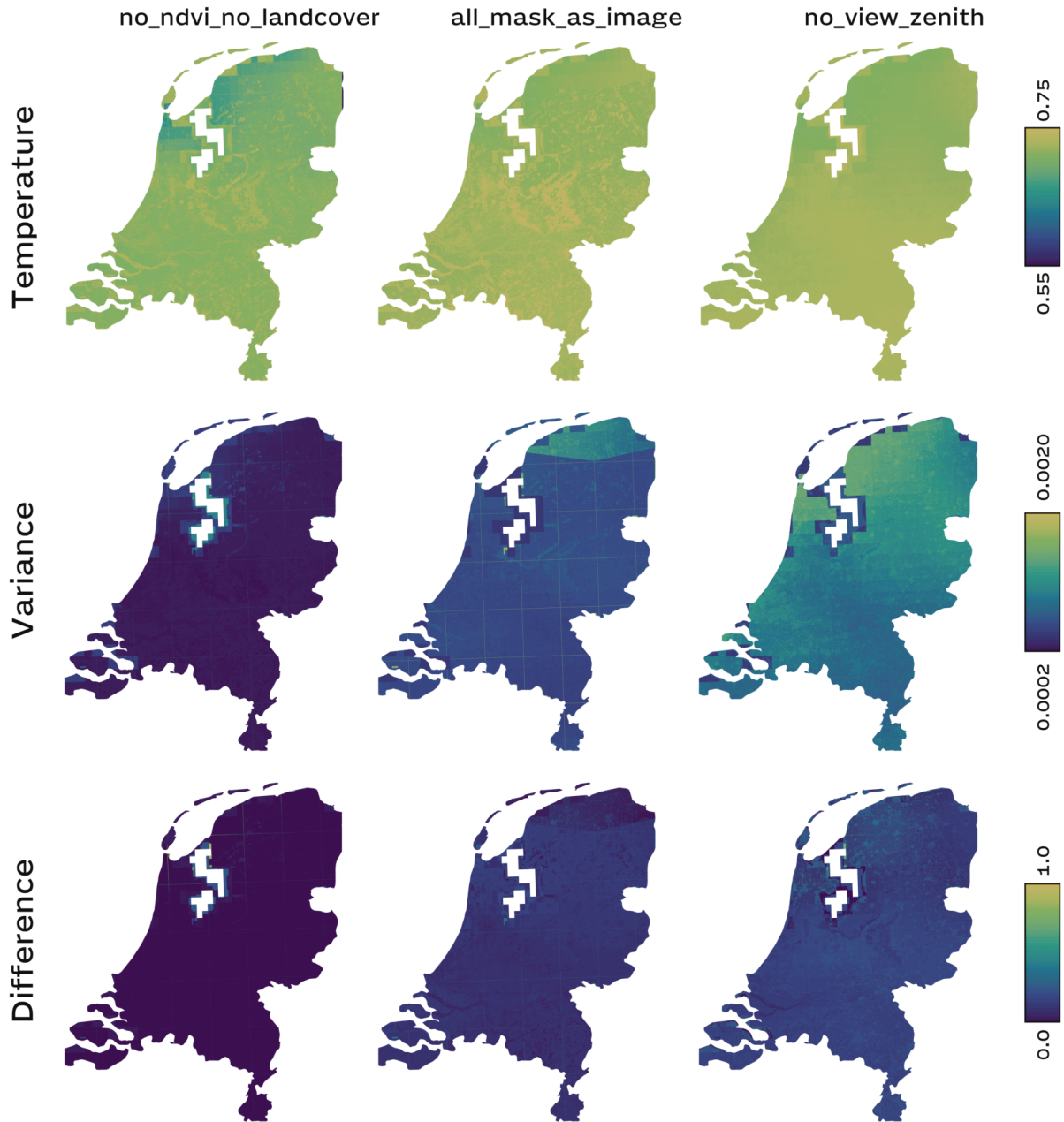


Figure 19: Temperature, variance and the difference between the two, for the three best performing models.

The variances shown in Figure 19 are visualized as a histogram in Figure 17. Interesting is that the “no_ndvi_no_landcover” model was extremely confident in its predictions when compared to the other two models, yet suffered the most from a correlation between the temperature and variance. Whether the mean and spread of variance values are connected to the correlation discussed above, or if they are both products of some other process is unknown.

Figure 18 shows the variances recorded during model training. When compared to the inference variances it is clear that the model training behavior with regards to variance is much more uniform. The “no_ndvi_no_landcover” became more confident in its predictions during inference, whilst “no_view_zenith” did the opposite. “all_image_as_mask” remained the most stable of the three, becoming less confident overall but not by much.

6.1 Performance between urban and non-urban areas

6.1.1 Method 1: To determine the performance of the model across different urban environments, stations were clustered together based on an estimate of the level of urbanization around them. For each station position, the NDVI and Landcover bands were sampled with a 32x32 pixel window. For NDVI samples, the mean of the returned values was calculated. For the Landcover samples the number of occurrences of the “not vegetated” class were counted, and this value was then divided by the number of pixels in the sampling window. Figure 20 shows how these values are correlated with each other. This figure also indicates into which cluster each point was added. Group 0 had 48 members, group 1 had 114 and group 2 had 107.

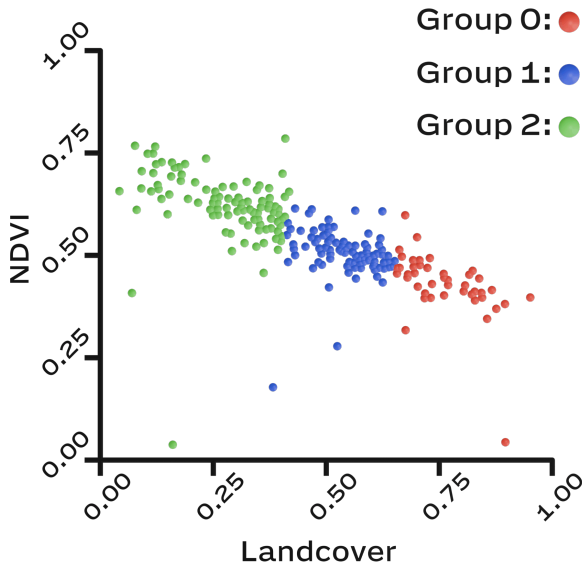


Figure 20: Graph showcasing the correlation between the NDVI band and the “non vegetated” class in the Landcover band. The different colors denote the groups created by K-means clustering.

For each cluster the mean loss of its stations was calculated. This process was executed for the 3 best performing models: “all_images_as_mask”, “no_ndvi_no_landcover” and “no_view_zenith”, resulting in a total of 9 values. To test if these values were statistically relevant, the stations were randomly divided into groups of equal size as the clusters. The means and standard deviations of these groups were then calculated. This procedure was performed 100 times and the results were averaged together. The calculated values can be seen in Table 2.

model name	cluster	cluster mean	sampled mean	sampled std
NNNL	0	-3.49	-3.41	0.39
	1	-3.32	-3.39	0.45
	2	-3.42	-3.39	0.47
AIAM	0	-3.51	-3.39	0.47
	1	-3.33	-3.38	0.50
	2	-3.39	-3.39	0.50
NVZ	0	-3.43	-3.33	0.49
	1	-3.25	-3.31	0.58
	2	-3.33	-3.31	0.56

Table 2: Table showing the results of the clustering and random sampling processes. Model names were abbreviated: NNNL: “no_ndvi_no_landcover”, AIAM: “all_images_as_mask”, NVZ: “no_view_zenith”

Although the results in Table 2 show that the cluster means deviate from the sampled means, they do not do so to the degree that indicates a measurable effect. Changing the sampling window size and number of clusters has an impact on the resulting means. However repeated testing showed that a window size of 32 and a cluster count of 3 (of which the results are shown above) produced the largest deviation from the sampled mean.

6.1.2 Method 2: Another method involves comparing the variance values calculated during inference. If either urban or non-urban areas are harder to accurately predict temperatures for, this will be reflected in the variance values.

The Landcover and NDVI bands were used to create a mask to approximate the position of urban areas in the Netherlands. For the Landcover band the “non vegetated” class was used to form a mask. In the NDVI band, values below 0.3 determined the formation of the mask. These intermediate masks and the result of their intersection can be seen in Figure 21.

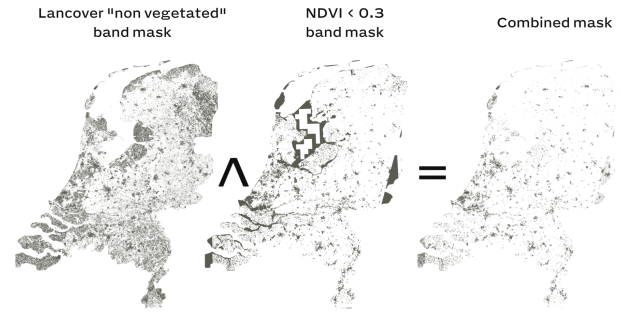


Figure 21: The two intermediate masks and the final combined “urban area” mask.

Values for the mean masked variance and mean total variance can be seen in Table 3. These results were calculated using the hour 1 variance band. Since the masked mean and total mean are nearly identical, it can be safely assumed that either the variance is a bad indicator of model performance or that the model performs equally well regardless of if an area is urban or non-urban. Changing the cutoff value for the NDVI mask drastically alters its shape, however testing indicated that means resulting from higher and lower values did not alter the results in any significant way.

model name	masked mean	total mean	total std
NNNL	0.000285	0.000285	0.000110
AIAM	0.000598	0.000597	0.000164
NVZ	0.000968	0.000966	0.000240

Table 3: Table showing the results of averaging the masked vs total variances. Model names were abbreviated: NNNL: “no_ndvi_no_landcover”, AIAM: “all_images_as_mask”, NVZ: “no_view_zenith”

6.1.3 Method 3: A final method of determining performance in urban areas involved calculating the standard deviation of the loss for each station, and then creating a raster by interpolating the standard deviation values. The result of this process can be seen in Figure 22. The interpolation was made using an inverse distance weighted algorithm. In this image brighter spots represent stations with a high standard deviation, while darker spots mean the opposite. At a glance there are no noticeable patterns. Factors such as proximity to water also do not appear to be affecting the standard deviation values.

Standard deviation

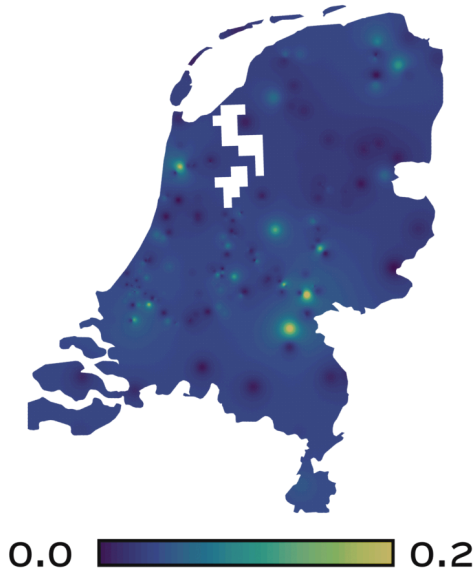


Figure 22: Interpolated raster of station loss standard deviation values. Bright spots signify stations with higher standard deviations, whilst darker spots denote the opposite.

Figure 23 provides a zoomed-in look at the city of Delft. Delft was selected as it has a high station density. Visible is how stations with good performance and stations with bad performance sit in close proximity to each other, which suggests that their performance is not related to the urban environment they reside in. If this were true, one would expect the pictured stations to have a more uniform appearance.

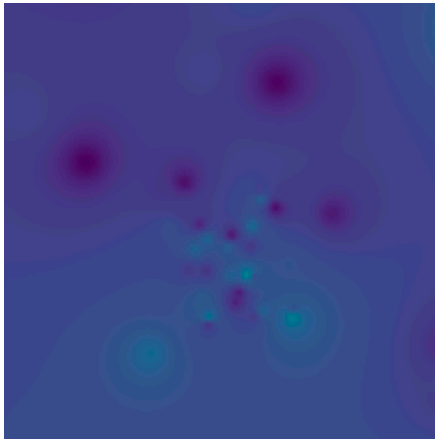


Figure 23: A zoomed in section of Figure 22. The area in question is the city of Delft. The image represents a 13km by 13km area.

7. Conclusion

Accurate and dense air temperature estimation can help identify areas in urban environments that suffer from, or are at risk of UHI. Previous work has indicated that spatio-temporal transformers are capable of handling the varied and complex data necessary for this task. This study describes and evaluates a spatio-temporal transformer model with integrated uncertainty predictions that provide valuable context. The model performs well during testing, and the outputs of the inference process show promise.

Unfortunately, predicted variances are partially correlated to air temperature estimations, indicating the model architecture or loss function are flawed. Due to time constraints k-folds cross validation was not utilized during training, making the outcomes of model training less reliable.

Further work should start with discovering why predictions and variances are correlated. Additionally, a greater focus should be put on the estimation of temperatures in urban environments. Development of efficient tools for the collection of data needed to train the model will help expand the models range beyond the borders of the Netherlands.

This study has demonstrated that spatio-temporal transformers have potential for producing large-scale air temperature predictions with estimated variances. It has also been shown that the models can be adapted to work with a variety of input data. This flexibility is valuable, as it allows for usage in areas that may not have a wide range of high quality remotely sensed data available. Such locations are also typically less well equipped to combat climate change and UHI.

8. Discussion

8.1 Ancillary bands

Clearly visible in the various inference images (such as Figure 19) are inland bodies of water with a predicted air temperature. These should have been set to nodata, as was done for the oceans.

8.2 Seasonal

Figure 9 and Figure 10 indicate that, with one exception, all time periods fall within a single standard deviation of the mean or are obvious outliers. It seems safe to assume that the models generalize well across both yearly and daily patterns.

8.3 Temporal

There are good indications that providing the model with information of previous captures helps improve performance. This is implied by the order of models in Figure 8, wherein the “base” model outperforms the “temporal_stack_0” model. Additionally, since the “temporal_stack_2” model placed above “temporal_stack_0”, it can be assumed that the extra temporal information provided is assisting the model more than the increased space consumed is costing it. Ultimately, it is surprising that incorporating information that is on average 143 hours old still provides tangible benefits to model performance.

Somewhat disappointing is how much the model relies on the time feature. Figure 8 shows beyond a shadow of a doubt that without the time feature, predictions become completely unreliable. This combined with the fact that the wind and SVF feature seemingly contribute nothing to performance, gives the impression that the feature transformer is exclusively occupied with processing the current date and time. This is in stark contrast to the temporal image embeddings, which was not able to contribute to increased model performance. Even when the time feature is omitted, the model is unable to properly parse the temporal embeddings, despite containing essentially the information.

8.4 Urban

The results of the three methods described in Section 6.1 make clear that there is no difference in model performance between urban and non-urban areas. It can therefore be assumed that the models generalize well for different areas.

8.5 Variance predictions

The predicted variances provide a lot of valuable context to the inference predictions. However, the current variance values are most likely correlated with the temperature estimations in some way, and therefore difficult to interpret. Additionally the behavior of the variances changes radically depending on the model used for inference, as seen in Figure 19, Figure 17 and Figure 18. Fully grasping the behavior of the variances is complex, as it seems to change from model to model. The fact that the variance histograms are so different between training and inference suggests that the model is being confronted with data it is wholly unfamiliar with. And logically speaking this makes sense. Since the training is fixed to 269 station positions, it is a certainty that the model is receiving inputs that are biased in some way. Specific bands are more susceptible to this, such as height, which is unlikely to change in any significant way over a four year period. It is understandable that a model has trouble interpreting 7.5 million new height samples when it has only ever been trained on approximately 200.

8.6 Temperature predictions

The predicted inference temperatures are promising. Estimated values make sense for the period that predictions are being made for. Figure 15 and Figure 16 show clearly that the model is capable of looking a number of hours into the future. Something concerning is how different the predicted values can be amongst models of similar test score, as indicated in Figure 19. The temperature predictions for the “no_view_zenith” model are strangely smooth.

8.7 Training

As mentioned in the conclusion, k-folds validation should have been implemented. This was skipped due to time constraints, as it would have quintupled the needed training time.

At least three additional model tests should have been run:

- A variant of “all_image_as_mask”, but with the QC and LST_err bands omitted.
- A variant of “all_image_as_mask”, but temporal embeddings disabled.
- A variant of “base” with both temporal embeddings and the time feature turned off.

Another option that would have been worth exploring was randomly dropping pixels in training input images. This might have led to models more resistant to unreliable data. Image transforms on data instances might also have been interesting, whereby the images are rotated / flipped, so as to simulate more training data.

The majority of tested models focussed on image data or SWIN parameters instead of the feature, cross attention, and linear layer parameters. With more time it would have been nice to explore the impact of the parameters associated with these other layers.

Acknowledgements

I would like to thank my head supervisor Dr. Azarakhsh Rafiee for her invaluable advice and guidance, and my co-supervisor Dr. Roderik Lindenbergh for providing important feedback and information.

Justin Schembri was instrumental in setting up the high-speed model training in the BK VRLab, which was graciously made available for usage by Hans Hoogenboom and Dr. Stefan van der Spek.

I am extremely grateful for the constant support and encouragement from my parents Amy Ramsey and Erik van Blokland, and my Brothers Paul and Benjamin. Benjamin also helped with downloading KNMI WOW data.

A special thanks goes out to my Geomatics peers: Alexandre Bry, Mingjie Teo, Phuong “Alena” Anh Ho, Carlo Cordes, Andika Hadi, Carmem Félix Aires, Ming-Chieh “Jerry” Hu, Stepan Prikazchikov, Frederick Auer and Juan Garcia Mezo for their companionship and conversation. Carlo also assisted in acquisition of the ERA 5 dataset, and Alexandre made the wonderful Typst template that this document is written with.

References

- Bazi, Y., Bashmal, L., Rahhal, M.M.A., Dayil, R.A., Ajlan, N.A., 2021. Vision transformers for remote sensing image classification. *Remote Sensing*, 13(3), 516.
- Benali, A., Carvalho, A., Nunes, J.P., Carvalhais, N., Santos, A., 2012. Estimating air surface temperature in Portugal using MODIS LST data. *Remote sensing of environment*, 124, 108–121.
- Bigg, G.R., 1991. Kriging and intraregional rainfall variability in England. *International Journal of Climatology*, 11(6), 663–675.
- Cawse-Nicholson, K., Johnson, M., Halverson, G., 2023. ECOSTRESS Collection 2 Level-1-2-3-4 Gridded and Tiled Data Products User Guide. https://lpdaac.usgs.gov/documents/1655/ECO_L1C-4_Grid_Tile_User_Guide_V2.pdf.
- Chen, H., Qi, Z., Shi, Z., 2021. Remote sensing image change detection with transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 60, 1–14.
- Copernicus, 2020a. Normalised Difference Vegetation Index 2020-2025. <https://land.copernicus.eu/en/products/vegetation/normalised-difference-vegetation-index-v2-0-300m>.
- Copernicus, 2020b. Land Cover 2020. <https://land.copernicus.eu/en/products/global-dynamic-land-cover/land-cover-2020-raster-10-m-global-annual#download>.
- Deilami, K., Kamruzzaman, M., Liu, Y., 2018. Urban heat island effect: A systematic review of spatio-temporal factors, data, methods, and mitigation measures. *International journal of applied earth observation and geoinformation*, 67, 30–42.
- Gillies, S., others, 2013. Rasterio: geospatial raster I/O for Python programmers. <https://github.com/rasterio/rasterio>.
- He, X., Zhou, Y., Zhao, J., Zhang, D., Yao, R., Xue, Y., 2022. Swin transformer embedding UNet for remote sensing image semantic segmentation. *IEEE transactions on geoscience and remote sensing*, 60, 1–15.
- Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., others, 2023. ERA5 hourly data on single levels from 1940 to present. *Copernicus climate change service (c3s) climate data store (cds)*, 10(1.24381), 24381.
- Hulley, G., Freepartner, R., 2019. ECOsystem Spaceborne Thermal Radiometer Experiment on Space Station (ECOSTRESS) Mission. https://lpdaac.usgs.gov/documents/423/ECO2_User_Guide_V1.pdf.
- Jiang, B., Wang, Z., Wang, X., Zhang, Z., Chen, L., Wang, X., Luo, B., 2023. VcT: Visual change transformer for remote sensing image change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 61, 1–14.

- Kawashima, S., Ishida, T., Minomura, M., Miwa, T., 2000. Relations between surface temperature and air temperature on a local scale during winter nights. *Journal of Applied Meteorology*, 39(9), 1570–1579.
- Kim, H.H., 1992. Urban heat island. *International Journal of Remote Sensing*, 13(12), 2319–2336.
- KNMI, 2022. KNMI data platform. <https://www.knmi.nl/nederland-nu/klimatologie/maand-en-seizoensoverzichten/2022/augustus>.
- KNMI, 2026. KNMI data platform. <https://dataplatform.knmi.nl/about>.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.
- NASA, 2018. ECOSTRESS remote sensing product. <https://ecostress.jpl.nasa.gov/>.
- Naserikia, M., Hart, M.A., Nazarian, N., Bechtel, B., Lipson, M., Nice, K.A., 2023. Land surface and air temperature dynamics: The role of urban form and seasonality. *Science of The Total Environment*, 905, 167306.
- Nix, D.A., Weigend, A.S., 1994. Estimating the mean and variance of the target probability distribution. *Proceedings of 1994 IEEE International Conference on Neural Networks (Icnn'94)*, 55–60.
- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in PyTorch. *NIPS-W*.
- Rafiee, 2026. Urban Thermal Modelling: A Spatio-Temporal Uncertainty-Integrated Transformer Framework for Urban Air Temperature Prediction Using Satellite Thermal Data and In-situ Sensor Measurements, Under review.
- Roy, S.K., Deria, A., Hong, D., Rasti, B., Plaza, A., Chanussot, J., 2023. Multimodal fusion transformer for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 61, 1–20.
- Salomonson, V.V., Barnes, W., Maymon, P.W., Montgomery, H.E., Ostrow, H., 1989. MODIS: Advanced facility instrument for studies of the Earth as a system. *IEEE Transactions on geoscience and remote sensing*, 27(2), 145–153.
- Svensson, M.K., 2004. Sky view factor analysis—implications for urban air temperature differences. *Meteorological applications*, 11(3), 201–211.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Weng, Q., Fu, P., Gao, F., 2014. Generating daily land surface temperature at Landsat resolution by fusing Landsat and MODIS data. *Remote sensing of environment*, 145, 55–67.
- WOW, K., 2026. KNMI WOW community data platform. <https://www.knmi.nl/kennis-en-datacentrum/uitleg/wow-nl>.
- Yan, H., Fan, S., Guo, C., Wu, F., Zhang, N., Dong, L., 2014. Assessing the effects of landscape design parameters on intra-urban air temperature variability: The case of Beijing, China. *Building and environment*, 76, 44–53.
- Yoo, C., Im, J., Park, S., Quackenbush, L.J., 2018. Estimation of daily maximum and minimum air temperatures in urban landscapes using MODIS time series satellite data. *ISPRS journal of photogrammetry and remote sensing*, 137, 149–162.
- Zhang, C., Wang, L., Cheng, S., Li, Y., 2022. SwinSUNet: Pure transformer network for remote sensing image change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60, 1–13.

Page left intentionally Blank

4. Closing discussion

Although the focus of this thesis concerns the design and testing of a machine learning model, the majority of work was spent acquiring, processing and managing datasets. This involved performing spatial queries to calculate only the exact subset of data needed, maintaining databases with sensor measurements and devising methods for matching thousands of image samples with their temporal neighbors.

One notable example is needing to build a web scraper so as to be able to download a vital dataset. This dataset was actively being shut down by its owners and no longer had a functional API. Fortunately, small samples of data could be downloaded via their website. The aforementioned scraper then spent three weeks downloading 3GBs worth of data, one file at a time.

Sampling raster files at specific positions and resolutions was necessary for both creating training data and performing inference. This required a good understanding of handling different CRSs and dealing with transforms and re-projections. Appreciation for good metadata was further strengthened when having to both read and write complex Geotiffs.

The Geomatics courses whose material and assignments shaped the knowledge needed for this thesis include:

- GEO_1001 Sensing Technologies: For an understanding of remote sensing fundamentals, both practical and theoretical.
- GEO_1002 GIS: For all the work done visualizing rasters and performing operations with them.
- GEO_1003 Positioning: For an understanding and appreciation of coordinate reference systems.
- GEO_1006 Databases: For all the practice with querying databases and the theory on efficient data usage.
- GEO_5019 Studio: For an introduction to modern geo technologies.

The mindset of all educators involved in the Geomatics program was also instrumental in shaping an environment that fostered creativity and cooperation. Most of the figures in this thesis were created with Blender, a practice that was honed and encouraged during the many assignments made.

4.1 Future work

The following is a list of experiments, alterations and additions that would have been interesting to pursue, had there been time:

- k-folds validation for model training would have lent greater certainty to the conclusions drawn.
- Additional model setup tests that could have further strengthened the results, or pursued higher model performance. They are mentioned in the discussion in part 3.
- A web-app to view the model descriptor and performance JSON files. Descriptor files defined how models were trained, and performance files recorded the progress of a model during training. An interactive webpage where these files could be compared against one another would have made model results easier to parse.
- Expanding the training dataset with data from outside of the Netherlands. It would have been interesting to see how the model performed when confronted with data from radically different climates and environments.
- A better system for the acquisition and maintenance of training data. The process for procuring data is still very hands-on. Building a system that could automatically fetch data for a specific region, or even update datasets when new ECOSTRESS captures become available would have been a fun challenge.

5. Reproducibility

5.1 Data

This thesis used data from four sources:

1. LST and its ancillary bands from ECOSTRESS (NASA, 2018)
2. NDVI and Landcover from ESA (Copernicus, 2020a, 2020b)
3. Sensor Air temperature, wind speed and wind direction data, and raster SVF imagery from the Dutch weather service: KNMI (KNMI, 2026)
4. Wind raster data from ERA-5 (Hersbach et al., 2023)

Of these four, 1, 2 and 4 are available globally and for free, and should therefore not be difficult to acquire for other regions. Countries other than the Netherlands also have free API access to their sensor measurements, but it is unknown how many there are and to what degree their methods and available data overlap with those of the KNMI. Although no tests have been performed regarding the performance of the model when only exposed to training data for a specific type of area, it seems safe to assume that a wide variety is necessary. The effectiveness of the model when applied to other regions can therefore depend on the coverage of stations. Related to this is the unknown impact that utilizing the Netherlands as a study area has had on drawn conclusions. It is possible that additional data types are necessary for the model to function properly in regions that do not resemble the features found in the Netherlands.

The model has proven remarkably flexible in terms of its provided input. Only LST imagery and air temperature readings are required data types. Users can append as much or as little additional data as they wish, adjusting to available computing power and data.

5.2 Code

All code for this thesis was created in Python, and dependencies were kept to a minimum to make usage easier. Code can be split into three main categories:

- Programs for the downloading of API data.
- Programs for the processing of input data.
- Programs related to describing the model, running training and inference, and preparing data.

Included in the project is a JSON pathing file. Users can edit this file with a number of local paths, and all scripts in the project will then be able to function properly.

Model setups are defined in JSON files. Editing these files alters how model training is executed. Additional datasets can easily be integrated by adding their information to these files and enabling their inclusion.

Code that was copied from online sources is noted as such, and is accompanied by a link to its origin.

6. AI usage

Generative AI was not used at any point in the making of this thesis. All images, text, code, references and mistakes were made by a human. This is also true for all the logical thinking and reasoning performed while working on aspects of this project. All received advice and help was of human origin.

References

Copernicus, 2020a. Normalised Difference Vegetation Index 2020-2025. <https://land.copernicus.eu/en/products/vegetation/normalised-difference-vegetation-index-v2-0-300m>.

Copernicus, 2020b. Land Cover 2020. <https://land.copernicus.eu/en/products/global-dynamic-land-cover/land-cover-2020-raster-10-m-global-annual#download>.

Cresswell, M., Morse, A., Thomson, M., Connor, S., 1999. Estimating surface air temperatures, from Meteosat land surface temperatures, using an empirical solar zenith angle model. *International Journal of Remote Sensing*, 20(6), 1125–1132.

Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., others, 2023. ERA5 hourly data on single levels from 1940 to present. *Copernicus climate change service (c3s) climate data store (cds)*, 10(1.24381), 24381.

Kawashima, S., Ishida, T., Minomura, M., Miwa, T., 2000. Relations between surface temperature and air temperature on a local scale during winter nights. *Journal of Applied Meteorology*, 39(9), 1570–1579.

Kim, H.H., 1992. Urban heat island. *International Journal of Remote Sensing*, 13(12), 2319–2336.

KNMI, 2026. KNMI data platform. <https://dataplatfom.knmi.nl/about>.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

NASA, 2018. ECOSTRESS remote sensing product. <https://ecostress.jpl.nasa.gov/>.

Prihodko, L., Goward, S.N., 1997. Estimation of air temperature from remotely sensed surface observations. *Remote sensing of environment*, 60(3), 335–346.

Salomonson, V.V., Barnes, W., Maymon, P.W., Montgomery, H.E., Ostrow, H., 1989. MODIS: Advanced facility instrument for studies of the Earth as a system. *IEEE Transactions on geoscience and remote sensing*, 27(2), 145–153.

Shen, H., Jiang, Y., Li, T., Cheng, Q., Zeng, C., Zhang, L., 2020. Deep learning-based air temperature mapping by fusing remote sensing, station, simulation and socioeconomic data. *Remote Sensing of Environment*, 240, 111692.

Sun, Y.-J., Wang, J.-F., Zhang, R.-H., Gillies, R., Xue, Y., Bo, Y.-C., 2005. Air temperature retrieval from remote sensing data based on thermodynamics. *Theoretical and applied climatology*, 80(1), 37–48.

Svensson, M.K., 2004. Sky view factor analysis—implications for urban air temperature differences. *Meteorological applications*, 11(3), 201–211.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*, 30.