High Resolution Temperature Prediction

Machine Learning Workshop : Project Booklet

Omri Maoz, Dan Barouch

Tel-Aviv University

{omrimaoz, danbarouch}@mail.tau.ac.il

Abstract

In this project we set to use machine learning (specifically deep learning neural networks) tools to predict and generate temperature (IR) images of different environment areas from input images taken by relatively cheap and available camera.

Our main approach was convolution network as the base asumption was that there is a natural function that determain temperature given a certain data (time, shadow, humidity etc.) and that temperature is dependence with the sorrunding area temperature . Given this assumption, we tried to predict each pixel temperature from its surrounding area to build the whole image rather than generating the entire image altogether.

1. Introduction

Infrared (IR) thermography is a process where a thermal camera captures and creates an image of an object by using infrared radiation emitted from the object. Thermal Imaging is used for many industrial applications that affect our daily lives. For example, Health services, Gas detection, Agriculture, Security, etc.

Environment scientists go out to nature and fly drones equipped with cameras to capture different source of imaging in order to explore different aspect of the environment. A thermal camera is an expensive tool compared to other technologies' cameras, which raise a need to generate and compute a thermal imaging from different imaging sources.

In this project, we were challenged to find a robust automated machine learning algorithm to predict and generate IR image of an environment area from few registers images of different sources of the same area. A register image set is a set of images of different sources transformed to the same coordinate system. The dataset and input data for this project were a few registers image set of 2 desert areas in the Negev desert, Israel, in high resolution, followed with a weather station data located somewhere within the image set captured area.

Deep learning tools and methodology would be the best approached for this challenge. A lot of work and study have been done using deep learning tools with thermal imaging to fulfill different tasks with good results. We chose to explore different CNN models, as they proved to achieve best results in analyzing visual imagery and expressing environmental effects. Moreover, to generate IR image we try to predict each pixel of the image by looking on fix size frame around the pixel of the input images, that's mean we treat it as a regression or classification problem.

Our code is available at https://github.com/omrimaoz/High_Resolution_Temperature_Prediction

2. Related Work

It has been shown that regression and classifications tasks from images are best achieved through CNN models.

Regression is a technique for investigating the relationships between variables and an outcome. Classification is the process of classifying or categorizing data points into predefined groups.

Till today, independent researchers and commercial hi-tech companies have created and trained state of the art, CNN based models, with which they achieved astounding results and managed to successfully complete new difficult tasks.

3. Our Approach

3.1 Datasets

Our dataset consists of seven register image sets, each contains an RGB and DSM (Digital Surface Model) of the same area and time. For each register image set using a deterministic algorithm, six preprocess images of different terrain characteristics are created:

- Height - The surface sea level.
- Slope - The angle of the slope of the surface.
- Real Solar - The intensity of solar radiation heating the surface.
- Shade - Whether the surface is shadowed.
- Sky View – The percentage of the sky that can be seen in that pixel.
- TGI (Triangular Greenness Index) - A vegetation index represents the greenness of the surface.

In addition, each image set has environment measurements from a weather station located within the image area.

Our focus on this project was on two areas, both in the Negev desert (desert characteristics) – Zeelim and Mishmar Hanegev.

Each image dimension is 1000×1000 pixels. We created the model dataset by sampling pixels or frames from an image set with different sampling methods.

3.2 The Baseline

For our baseline, we built a two straight-forward neural network models to predict a Pixel-to-Pixel IR image. This means that we tried to find a function to predict an IR pixel from its corresponding input images pixels.

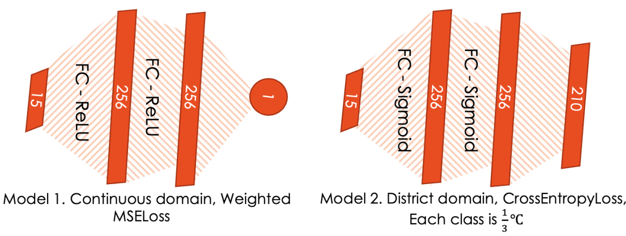


Fig 1. 2 neural network models architecture of the baseline

Model 1 classified with regression and Model 2 classified to a district temperature domain.

Model	Acc $\frac{1}{2}$°C	Acc 1°C	Acc 2°C	MAE	MSE	Acc $\frac{1}{2}$°C, Entire 7th image-set prediction
Model 1	41.2%	61.6%	87.6%	0.879	1.499	64.1%
Model 2	53.8%	73.8%	94.6%	0.632	0.813	67.2%

Table 1. 2 baseline model evaluations and results

When looking at the results we see that the discrete model performed slightly better during training and when generating a new IR image.

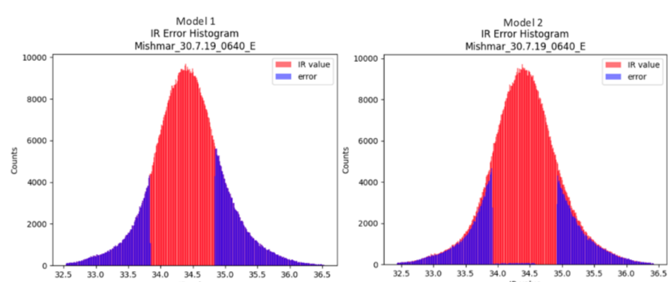


Fig 2. 2 baseline model error histograms

By examining the error histogram for each model, we see that the two models mostly learn the average temperature of the images rather than the correct IR value of pixels. The discrete model performed better regarding this issue.

Moreover, the predicted IR images produced from both models were random noises. In this baseline we yet to consider the effect of neighbors' pixels on the pixel temperature which we thought will help to achieve better results.

3.3 From baseline to a working model

To test the hypothesis from the baseline section, we tried to build and use CNN models. Instead of Pixel-to-Pixel prediction, the new model is Frame-to-Pixel prediction, which means that we sample a square frame around the pixel that we want to predict, with a fix window-size that we tested and optimized to the image's resolution.

We suggested 3 CNN models:

- ConvNet - A simple, relatively shallow customed CNN.
- ResNet18 not pretrained – This model work great on many computers vision tasks. We used the residual block to better train a deeper network. We chose 18 layers because our task does not require extracting feature necessarily, and we believe that this level of dept will work better.
- ResNet18 pretrained – With pretrained weights.

3.4 Relative Sampling

We saw in the baseline results and error histograms that the models tend to learn the average temperature of the image. By uniformly sampling pixels to predict from the training IR images, due to the normal distribution of the IR values we take mostly the pixels with IR value close to the average value. In order to compensate for that, we need to prioritize the less common IR value pixels. To do that we defined IR levels of 0.2℃ width each and distributed each pixel to its level, then we took equal number of pixels from each level. If one level had less pixels than the required amount, we took the difference from the closest level that has pixels left. In this way we sampled pixels for our training dataset and expressed all IR values in training.

3.5 IR bias

When we trained our models on six image sets and tried to predict the 7th IR image, we saw that the predicted IR image resembled its original but darker or lighter grayscale. When we examined their distributions, we saw a similar distribution differs in bias.

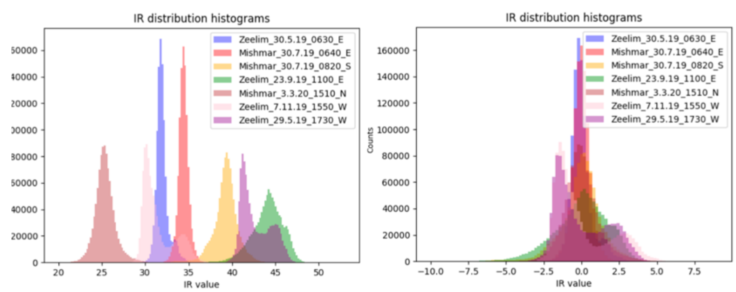


Fig 3. Left, IR distribution of the 7 images. Right, IR distribution of the 7 images after subtracting the mean of each IR image.

To solve the bias problem, we tried to predict only the offset of each IR pixel from the bias of the IR image which is the average temperature of the IR image. To test this hypothesis, we preformed two steps.

- First step - We computed each image set their average temperature and subtract it from the IR image as shown in Fig 3. That way, we omitted the bias effect and could better generalize to every IR image. We trained a model with unbiases labels. Then we produced an IR prediction image and restored the image real bias from the image set IR image.
- Second step - In production stage, a user will have only the input image set and the weather station data and will try to produce an IR

image using our model. The model will produce an unbiased IR prediction image and our algorithm will need to restore the image bias without knowing the bias from the real IR image. To do that, we suggest computing the bias using only the input data.

We first noticed that the mean temperature of each IR image is not necessarily the ground temperature measured in the weather station of the same area.

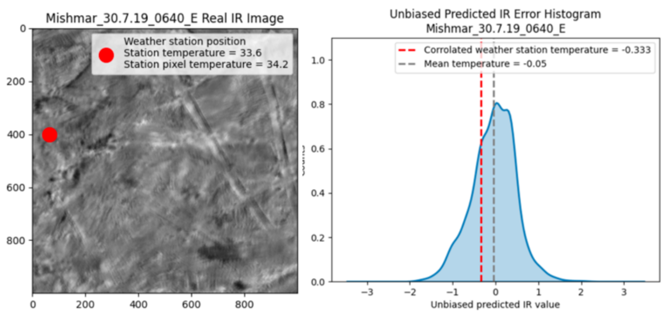


Fig 4. Left, true IR image, the station position, the temperature measured in the station and the pixel temperature of the station. Right, the unbiased IR prediction distribution, the mean temperature, and the correlated temperature of the station according to its position.

By knowing the weather station position in the image set area we computed the temperature difference of the weather station from the mean temperature on the predicted IR distribution.

The final IR prediction image will produce with the following equation:

$$Temp_{xy} = PredictedTemp_{xy} + StationTemp - (CorrolatedStationTemp - MeanTemp)$$

Equation 1. $\boldsymbol{PredictedTemp_{xy}}$ **- The predicted temperature of the model at pixel** (x, y). $\boldsymbol{StationTemp}$ **- The station measured temperature.** $\boldsymbol{CorrolatedStationTemp}$ **- The correlated station temperature on the predicted IR distribution according to its position.** $\boldsymbol{MeanTemp}$ **– The mean temperature of the predicted IR distribution.**

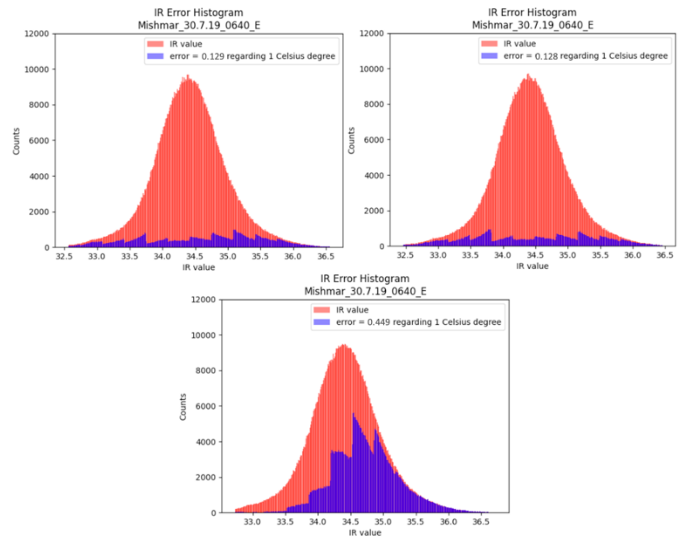


Fig 5. Top left, IR prediction with true bias. Top right, IR prediction with bias computed from station pixel temperature. Bottom, IR prediction with bias computed from station measured temperature.

One more point to notice is that the bias is not relevant for the IR prediction image visualization as the image need to be converted to grayscale and so only the relations between pixels matter. An example is presented in **Fig 6.** at the **Results** section.

4. Evaluation – Metrics

To evaluate the results of our model, we used Accuracy, MAE and MSE metrics.

Accuracy is defined at the percentage of pixels whose temperature the model was able to predict with a certain error range. We examined the accuracy with relation to an error range of 0.5°C, 1°C and 2°C in each direction (one dimension temperature scale), in order to understand how well the model performs given different temperature resolution requirements.

We chose the MAE (Mean Absolute Error) metric to evaluate how far the average pixel prediction is from the real temperature, and the MSE (Mean Square Error) to understand how well our model handles the less common, extreme temperature values, as those values are more likely to create large errors.

5. Results

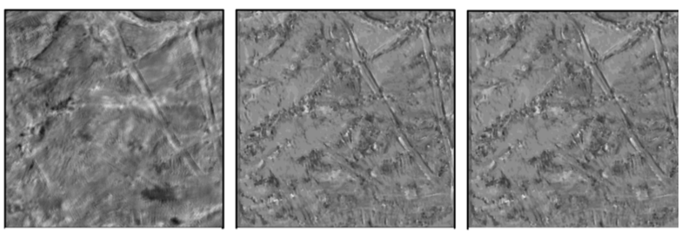
In the proposed approach, we trained and tested the three suggested models, with "relative" sampling and normalized the true bias.

Model	Acc $\frac{1}{2}$°C	Acc 1°C	Acc 2°C	MAE	MSE	Acc $\frac{1}{2}$°C, Entire 7th image-set prediction True bias	Acc $\frac{1}{2}$°C, Entire 7th image-set prediction Station bias
ConvNet	53.2%	70.9%	91.4%	0.71	1.175	44.5%	19.1%
ResNet18	87.2%	97.0%	99.8%	0.246	0.137	60.4%	17.6%
ResNet18 - pretrained	89.3%	98.1%	99.9%	0.231	0.122	58.2%	48.6%

Table 2. Results and evaluation of 3 suggested working models, "relative" sampling, trained on 6 images with true bias normalization.

Based on the table above, the ResNet18 with and without pretrained weights has shown the most promising results with slight advantage to the pretrained weights model.

ConvNet managed to achieve good results in earlier stages when trained on a single image set.



Mishmar_30.7.19_640_E Real IR image.

IR prediction image with true bias compute from real IR image.

IR prediction image with bias compute from the station measured temperature regarding the station location.

Fig 6. IR prediction image using our best ResNet18 with pretrained weights model.

To compare the "relative" sampling to uniformly sampling we used ResNet18 with pretrained weights and relatively small samples number from six image sets. Due to the nature of normal distribution, when sampling a lot of samples, without repeating, from the images distribution we collected all the extremist samples and left with no choice but to sample more data from the center of the distribution. By that we minimize the effect of the "relative" sampling.

Model	Acc $\frac{1}{2}$°C	Acc 1°C	Acc 2°C	MAE	MSE	Acc $\frac{1}{2}$°C, Entire 7th image-set prediction
ResNet18 - pretrained SFP	69.3%	84.5%	97.7%	0.498	0.661	57.1%
ResNet18 - pretrained RFP	66.8%	82.6%	95.5%	0.508	0.719	52.9%

Table 3. Results and evaluation of ResNet18 with pretrained weights using "relative" sampling (RFP) and uniform sampling (SFP).

Based on the table above, we see that the uniform sample method achieved better results. But when examining the error of the two methods we see that indeed the "relative" sampling method produced more balanced images and dealt with prediction the extremist temperature of the image set.

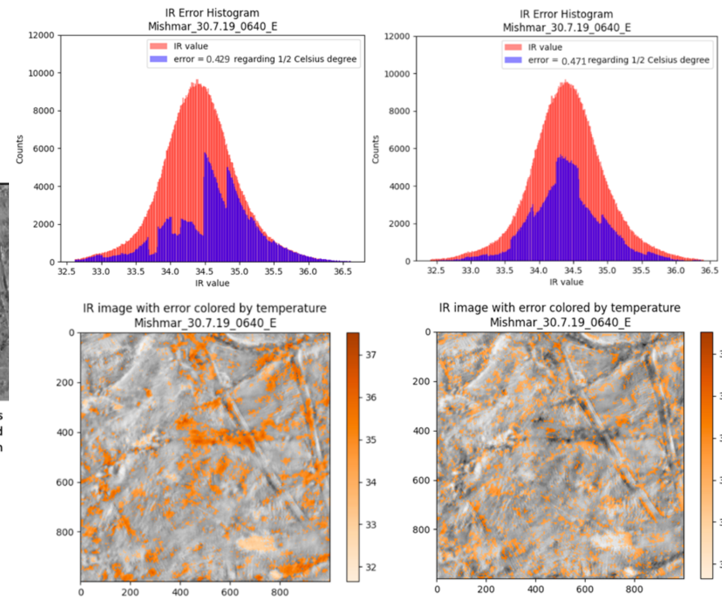


Fig 7. Top, IR prediction with error distribution. Bottom, IR prediction image with error colored by temperature. Left, uniform sampling method (SFP). Right, "relative" sampling method (RFP).

As we see on Fig 6. uniform sampling rarely corrects on extremist temperature and in the predicted image errors are darker or brighter. "Relative" sampling did manage to predict the extremist temperature but paid with more errors in the average temperature.

6. Summary and Conclusions

In this project, we explored and tested numerous deep learning approaches, algorithms, and models in order to make deep and comprehensive experiments and to best solve the problem at hand.

We started with the baseline, then developed the first working model for a single image set, adjusted hyperparameters, developed preprocessing methods and finally trained a deeper model on the whole dataset with great results.

Our model showed that we can train different area and environment properties such as time, season, etc. by applying bias normalization with which accurate measurement is approaching the true bias of the IR camera image.

Our main conclusions are:

- ResNet18 produces the best results. The state-of-the-art architecture allowed us to train on a deeper network and to better generalized and minimize errors. Moreover, we see that ResNet18 pretrained weights are redundant for this problem as we managed to approach the same results without those weights. An explanation for this could be that our data doesn't hold features resembling the features extracted from the massive dataset the model was trained on.
- We see that the bias plays a great role both in training and correcting the prediction. When correcting the prediction image using the true bias and even with the bias computed from the station pixel temperature, we achieved very good results, but small inaccuracy with the station measurement regarding the true temperature led to a large enough offset, and in turn to poor results.
- Uniform sampling achieves better results than "relative" sampling. The temperature is normal distributed and therefore when uniformly sampling it, most of the samples are around the average temperature. Then the model will learn to predict those temperature and will predict more pixels than the latter simply because there are more pixels there than in the extremist temperatures. But when producing prediction images, we see the foremost misses the extremist pixels and having more gathered error areas comparing to the latter which predicts the extremist temperatures and its errors are more uniform distributed, and therefore produces clearer image.

References

[1] Kaiming He. Xiangyu Zhang. Shaoqing Ren. Jian Sun. Deep Residual Learning for Image Recognition. *Computer Vision and Pattern Recognition* (2015).

[2] Yong Xian. Juan Su. Da Q. Zhang. Wei L. Guo. I-GANs for Infrered Image Generation. *Complexity* (2021).

[3] Vasilis Vryniotis. How to Train State-Of-The-Art Models Using TorchVision's Latest Primitives. *PyTorch* (2021).