Retrieval of Visible Spectrum Information from Thermogram images using Backpropagation network as the Kernel function

Prateek Durgapal
IIT2019014

Abstract - This study adopted a non-parametric functional form of the kernel to be utilized for the purpose of converting an infrared image to a grayscale one which is developed using a simple backpropagation network. The result of this study shows that the novel kernel-based approach not only can retrieve a sizable amount of visible spectrum information of the object but also may retrieve more in terms of 0.5 cm depth information in case of a medical thermogram of the human face. The thermogram to visible range image conversion model proposed in this work can work as a naive yet efficient kernel without the requirement of any prior knowledge of the thermal ambiance of the object.

Keywords - ANN, Backpropagation, Thermography, Kernel function, Non-parametric, Context-window, Grayscale

I. INTRODUCTION

This research requires us to find ways to obtain a grayscale image given a thermograph image. All objects emit electromagnetic radiation, primarily in the Infrared (IR) wavelength, which cannot be seen by the naked eye. However, IR radiation can be felt as heat on one's skin. The hotter the object, the more IR radiation it emits. IR images basically represent these heatmaps. Our method is based on a non-parametric backpropagation network as kernel function which is a novel method for the conversion of a thermal infrared image into a visible spectrum image.

Since the wavelength of IR is longer than Visible Spectrum (VIS), its resolution is very low comparatively. Because of this, VIS images contain much more information than the corresponding IR image.

Hence, the challenge is to somehow map this relatively little information (provided by the heatmap) into a high-dimensional feature space (VIS). And hence we use kernel functions.

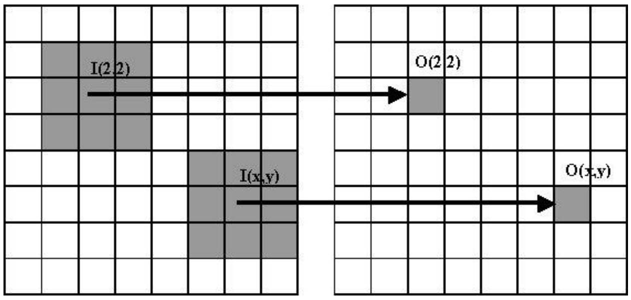

Fig. 1. Square context window (size = 3x3)

II. PREVIOUS WORK

Paper [1] deals with the approximate conversion of many-to-many mapping between VIS and IR images into a one-to-one mapping. The images are aligned, registered, and cut into small pieces. The linear regression is done on each piece, and pixels in different pieces are taken as independent, which effectively removes the nonlinear relation. Then, linear Canonical Correlation Analysis (CCA) is applied followed by Locally Linear Embedding (LLE) to recover the lost information.

Paper [2] is based on the idea of reconstruction of the VIS image from the given IR face image. Just like the previous method, two steps are used - Global Reconstruction and Local Refinement. Global Reconstruction involves a new thermal face image that will be reconstructed to a visible face image. This global image represents the globally reconstructed face image and undergoes enhancement later in the Local Refinement step. The Local Refinement step is where the details of the face are reintroduced, as it was lost in the Global Reconstruction. In order to accomplish this, multiple reconstructions to patches of the globally reconstructed image are employed.

Paper [3] produces a fully automatic method that makes use of Convolutional Neural Networks (CNNs) for IR to VIS transformation. Two approaches were used out of which the integrated approach (2nd) estimated both the luminance and the chrominance of the VIS image using

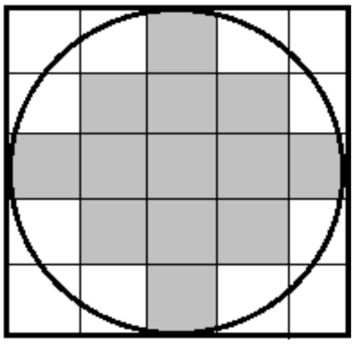

Fig. 2. Circular Context Window (diameter = 5)

the proposed architecture. The network has an encoder-decoder structure based on the generator architecture combined with skip connections. There were quite some failures in this approach. Firstly, the model fails to colorize the road markings since they have the same apparent temperature as the road paving. Secondly, the brake lights of the vehicles were not colorized correctly as turning on the brake light does not change the apparent temperature of the lamp cover.

IV. PROPOSED APPROACH

We approach this problem by creating the dataset in such a way that retains the local feature of a pixel.

A. Initial Data Acquisition

We have used the OSU Color-Thermal Database from OTCBVS Benchmark Dataset Collection. The details regarding the dataset are as follows:

- Number of images = 17,089
- Format of images =
  Thermal: 8-bit grayscale bitmap,
  Color: 24-bit color bitmap
- Image size = 320 x 240 pixels
- Sampling rate = approximately 30Hz

B. Pre-processing

First, we have converted the colored images into grayscale images. Then the corresponding (thermal, grayscale) image pairs are stored in the memory as a data frame. Then three variables as hyperparameters are created. The variables are:

- *perc_images*: which controls how many images to consider
- *perc_pixels*: which controls how many pixels per image are considered.
- *window_size*: the size of the context window taken.

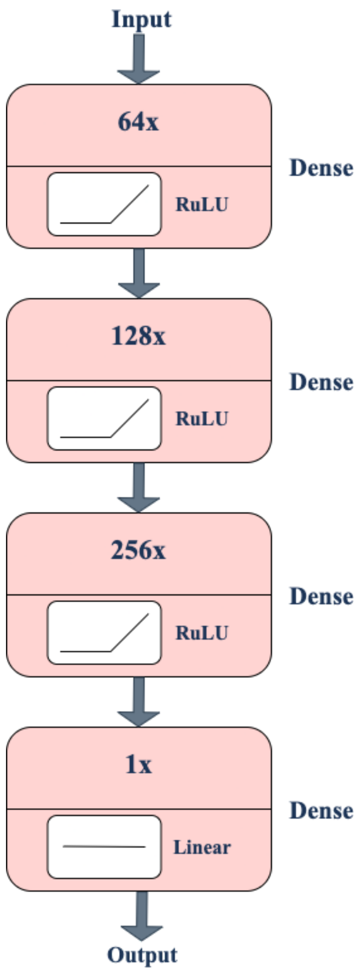

Fig. 3. Design of Neural Network Model

After this *perc_images* number of images are sampled from the dataset for computationally effective reasons.

C. Dataset Creation

First, an initial empty dataframe is created. Then we traverse each thermal image in the sampled dataframe one by one. Firstly both the images of the pair are normalised by using min-max normalization given by the formula:

$$I_{new} = (I - I_{min}) / (I_{max} - I_{min})$$

Here I_{new} is the new intensity value, I is the current, I_{min} is the minimum and I_{max} is the maximum intensity value in the image.

Then we take the difference of intensity values of the current image with the previous image. This allows us to focus on the pixel values which have undergone some

Fig. 4. A thermal image

change with respect to the previous image. The difference values which are greater than some threshold (chosen manually). Now for each of these selected pixels, we store this pixel as well as its neighboring pixels as defined by the context window shape and size. An example can be seen in Fig. 1 and Fig. 2 of square and circular context windows. Finally, we also append the corresponding grayscale pixel into the dataframe. For corner and edge pixels whose neighbors are outside the image, the intensity value of 0 has been appended. The final dimensions of the dataset have now become (*2834895*, $window\_size^2 + 1$).
After the completion of the above process, we divide the newly created dataset into training, validation, and testing data in the ratio 7:2:1.

D. Network Training
We have created an Artificial Neural Network (ANN) for this task. The architecture of this network can be seen in Fig. 3. Then we compile our model with loss chosen as

Fig. 5. A grayscale image

Fig. 6. The reconstructed image

root mean squared error (RMSE) and optimizer as Adam with default parameters. Before training, we also used some callbacks:

- Model Checkpoint: to store the best weights of the model whenever the validation loss improved.
- Early Stopping: Stop the training whenever the validation loss has stopped improving over a period of some epochs (20 in our case)

Now we train our model for 1000 epochs with a batch size of 32. The final training loss (RMSE) achieved was 0.0231 and the final validation loss was 0.0237. On evaluating the testing dataset, the testing loss was 0.0231 as well..

E. Reconstruction
We take a test image (shown in Fig. 4), apply min-max normalization again, and follow the same procedure as

Fig. 7. After applying high pass filter

Fig. 8. After applying Gaussian Blur

creating the dataset. For each pixel, we calculate its context window pixels and pass this data through the model for prediction of the corresponding grayscale pixel. We append the grayscale pixels to form our reconstructed image. The RMSE for this reconstructed image is 0.02275. The result could be seen in Fig. 6 alongside the grayscale image in Fig. 5.

F. Post-processing

Finally, we apply some image processing techniques to enhance the predicted image and reduce the RMSE even further. Firstly we apply Gaussian Blur (window size taken was 5 x 5) to the image, Fig. 9, to remove the additional noise from the picture. Next, we use a high pass filter (Laplacian in our case) on the thermal image.

Fig. 9. Final image after post-processing

The filter used was as shown below:

-1	-1	-1
-1	8	-1
-1	-1	-1

Lastly, we add the previous output to the blurred image to obtain the final image, Fig. 11. The final RMSE for this image is 0.02170.

VI. RESULTS AND DISCUSSION

Our neural network worked as we expected it to work and with the slight processing of the image produces better results. Table 1 highlights the results for different types of windows which we have tested and their corresponding error values.

Context Window Type	Error
Square context window (size = 5x5)	0.03591
Square context window (size = 7x7)	0.02275
Circular context window (diameter = 5)	0.0441
Circular context window (diameter = 7)	0.02978

Table 1. Comparison of context window performances

From these, we can observe that the square context window with size 7x7 has the best results.

REFERENCES

[1] M. Dou, C. Zhang, P. Hao, and J. Li, “Converting thermal infrared face images into normal gray-level images,” in ACCV, 2007.

[2] Kresnaraman, D. Deguchi, T. Takahashi, Y. Mekada, I. Ide and H. Murase, “Reconstructing Face Image from the Thermal Infrared Spectrum to the Visible Spectrum” in Sensors (Basel), 2016.

[3] A. Berg, J. Ahlberg, M. Felsberg, “Generating Visible Spectrum Images from Thermal Infrared” in IEEE/CVF, 2018.

[4] R. Usamentiaga, P. Venegas, J. Guerediaga, L. Vega, J. Molleda, F. Bulnes, "Infrared thermography for temperature measurement and non-destructive testing" in Sensors (Basel), 2014.

[5] M. Limmer, H.P.A. Lensch, Daimler AG, “Infrared Colorization Using Deep Convolutional Neural Networks”, arXiv:1604.02245, 2016.

[6] Z. Dong, S. Kamata, T.P.Breckon, “Infrared Image Colorization using a S-Shape Network” in 2018 25th IEEE International Conference on Image Processing.