Mapping RGB Image to Core Body Temperature and NIR Image Generation

Rituraj Kulshresth
Computer Science and Engineering
Indian Institute of Technology, Jodhpur
Jodhpur, India
kulshresth.1@iitj.ac.in

Saurabh Burewar
Computer Science and Engineering
Indian Institute of Technology, Jodhpur
Jodhpur, India
burewar.1@iitj.ac.in

Romi Banerjee
Computer Science and Engineering
Indian Institute of Technology, Jodhpur
Jodhpur, India
romibanerjee@iitj.ac.in

Debarati Bhunia Chakraborty
Computer Science and Engineering
Indian Institute of Technology, Jodhpur
Jodhpur, India
debarati@iitj.ac.in

***Abstract*—In this paper, we propose a possible method to map RGB face images to the core body temperature. We use convolutional neural networks to predict the core body temperature for a given RGB face image by training them on two different datasets. We also propose an experiment and a method to generate a NIR image from an RGB image. For this we use a Convolutional neural network to train the model to predict the pixel value for each NIR image by learning from the dataset of RGB - NIR pairs. Following the generation of a NIR image, it is processed to adjust the brightness and contrast for the shadows.After this the NIR image can be used to check the relation with the temperature of the body.**

I. Introduction

In these COVID times, as much as staying at home is important, it has also become very important to monitor people who step outside. The best way to recognize possible carriers is to measure body temperature. Normally, body temperature is something that would be taken by a thermometer. However a thermometer may not be present at all places at all times. Hence something much easier can be used in its place. Here we are trying to build a solution that could make this process easier.

The normal human body temperature range is typically stated as 36.5-37.5 °C (97.7-99.5 °F). However, human body temperature is variable and dependent upon one's sex, age, time of day, exertion level, health status (i.e. illness, menstrual cycle in females), the location in/on the body in which the measurement is being taken, the subject's state of consciousness (waking, sleeping or sedated), as well as emotional state. Body temperature is maintained within normal range by thermoregulation whereby the lowering or raising of temperature is triggered by the central nervous system.

Oral thermometers are an easy tool to measure the temperature of a body since they give a very accurate and convenient method to get the core temperature of the body. (For this report we are assuming that the core body temperature is measured most accurately using rectal temperature measurement). Oral thermometers report a temperature which is 1-2 degree lower than the core body temperature. Since using thermometers is invasive and requires contact with the body of a person it has to be avoided in the current times. Instead, an easier and quick solution is needed which can give accurate results quickly. We can use a thermal camera to get the accurate temperature of the surroundings and a person. However, a thermal camera is very costly. Using infrared thermometers gives a better solution to the problem. infrared thermometers are cheaper but may not be available to all the public due to their exclusivity. Therefore another approach is needed to find temperature non-invasively. Since each of us has a camera with us at all times we have tried to find the temperature of a body using a regular RGB camera. For this, we have experimented with convolutional neural networks

We have known for far too long that it is efficient and versatile to use IR radiations to measure the internal heat of a body and in turn, measure its temperature. This is possible due to the relation $E = h\nu$. The IR radiations are a good source of measuring the temperature since IR radiations range from approximately 700 nm to 1 mm. and are not easily disturbed by the intensity of the visible light. Hence they can be used in high light intensity regions too.

In a typical day-to-day use camera, a lens concentrates the light onto a sensor consisting of a lattice structure of chips sensitive to one or more spectral bands. For a daily use DSLR, this translates to around 350-750 nm. NIR ranges from the 650-2500 nm range (approximately) and also has a lot of advantages in spectroscopy and temperature measurement. In this paper, we try to find out if there is a possible way to get a NIR image from an RGB image from a simple daily use camera. This problem can be viewed as a Domain Adaptation or an Image Re-colorization problem. For this experiment, we will be using a convolutional neural network to train and perform a direct estimation of the intensity of each pixel. We use a set of RGB and NIR images that were captured using Nikon D90 and Canon T1i cameras, using B+W 486 (visible) and 093 (NIR) filters. The wavelength range of the

NIR images was 750 to 1100 nm. Extensive experimentation with the CNN models demonstrates a potential approach to get the NIR images.

II. Data collection

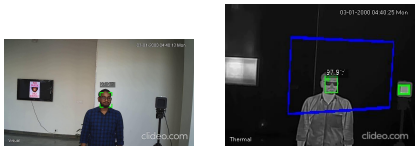
A. Dataset I

The first data was collected by taking RGB images of the forehead regions of the participants and taking their core body temperature using an IR thermometer/thermal gun in a room with artificial lighting. The images correspond to the temperatures 96 °F, 97 °F, 98 °F, 99 °F, and 100 °F. The temperature was measured for varying times of the day after various activities which were also included. However, due to the COVID restrictions, the data collection was limited to a few participants causing the data to be skewed towards the normal body temperature of about 97 / 98 °F which caused problems in training the model with the data.

B. EPFL dataset

This dataset contains a set of RGB and NIR images of different scenery ranging from forests and lakes to cities and buildings. The dataset consisted of 477 pairs of RGB and NIR images. The images were captured using separate exposures from modified SLR cameras, using visible and NIR filters. The cutoff wavelength between the two filters for RGB and NIR images is approximately 750nm.

C. IITJ Dataset



(a) (b)

Fig. 1: (a) RGB and (b) NIR image from IITJ dataset

The data was collected on the IITJ campus which includes RGB and thermal videos of participants using a thermal camera [Add details regarding thermal camera], their core body temperature, and factors that can affect the data such as ambient temperature, activity before collection. This data had temperatures in the 96 °F, 97 °F, and 98 °F range. The data presently being used still have posed the same problem that we encountered before and data from more participants on different conditions is necessary to balance the dataset.

III. Approach

The Initial approach of the experiment was to first find a relation between the temperature and the RGB images of the face of a person. Another variation of the experiment was to find a relation between the NIR images of the face and temperature. However, due to the availability of the data of face in the NIR spectrum, we considered an extension to the experiment by adding another part that is to generate NIR images from RGB images. For this experiment too we faced the same issue of data availability. Hence, we tried the experiment on the available NIR and RGB images to get an idea of whether NIR images could be generated from an RGB image and still have same brightness and contrast as the original image. the next part of the experiment will be to check if this generated image in NIR can be used in experimenting with the Temperature NIR relation.

A. RGB to Temperature Mapping

To find any relation between the RGB image of a person and the temperature of the person we have experimented with various convolutional neural networks to find any underlying mapping function.

1) Post processing: For dataset I, The images have been divided based on their respective body temperature. The data was resized to 256 x 256 pixels for easy training. Appropriate weights were added to each temperature class to adjust for the unequal distribution of data.

In case of the IITJ dataset, the data had temperatures in the 96 °F, 97 °F, and 98 °F range so training was also done in this range only. We used the RGB videos and extracted frames from the videos. Then the faces were detected and cropped out. These images were then sorted according to the temperature captured by the IR thermal gun.

2) Experiment: Since there was no previous research as to what kind of model would give proper results, We experimented with models consisting of 1 to 6 convolutional layers followed by 0 to 3 dense layers with layer sizes of 32, 64, 128 in all possible combinations.

B. RGB-NIR and NIR to Temperature Mapping

1) RGB-NIR Imaging: Each RGB image is made of a 3-dimensional array with each value denoting either of the R, G, B channels value for each pixel. The value of each pixel can range from 0 to 255 which is synonymous with the intensity of the brightness. The RGB images are captured by using a sensor for a particular wavelength range for each of R, G, and B. Typically the wavelength ranges of the channels are assumed to be around (in nanometer) Blue - [450 - 510], Green - [530 - 590], Red - [630 - 680] Similarly NIR images are captured using a sensor which starts near the R channels range and extends up to 1200 nm. NIR images can be single or multi-channeled, however, the multi-channeled ones have the same pixel values for all channels. The NIR images have the pixel value corresponding to the NIR range. The cumulative intensity of the range of wavelength is taken and gives a suitable value ranging from 0 to 255.

In order to generate the NIR images we first experimented with a Domain adaptation approach with the auto-encoder CNN. According to the result, we selected the models which gave us the best results. Then we tried an image re-colorization approach to again experiment with various CNNs. After we got sufficiently accurate results for the dataset, we used the same models to train on the RGB, NIR pairs of face data. We did this assuming that the initial models were able to learn every

color scheme from the variety of images provided by the EPFL dataset and find a probable relation to the NIR counterpart, and hence they would be able to get a similarly accurate result in the IITJ dataset of the faces.

This approach consists of three steps: First, preprocessing is performed to normalize images and augmenting the data. Second, a NIR image is inferred from the RGB input using CNN. Third, postprocessing is performed by filtering the output to compensate for the loss of details in the network.

2) Preprocessing: The preprocessing of the first dataset (EPFL) includes two components. The first is the reading and resizing of the image in a 1:1 ratio, in order to make training easier. The second is the augmentation of the data in order to increase the dataset. This includes rotating, flipping, and varying pixel brightness to create various variants of an image to increase the data for training.

The IITJ dataset consists of RGB and NIR videos but the NIR videos were of very low resolution. Moreover, there were inbuilt features of the camera device used that made face detection in the NIR images very difficult and inaccurate. So for the generation of NIR images from RGB, we used the whole frames of the videos. The dataset is preprocessed by scaling and cropping the RGB images in the Dataset to be equal in size and scale to the NIR images. After resizing the images for both RGB and NIR a new dataset was created.

3) Post-processing: The image generated by the convolutional network needs to adjusted and this is done using by first, filtering the image using a joint bilateral filter to reduce noise towards the edges and then adjusting the contrast to get better details in the shadows.

A joint bilateral filter is an edge-preserving and image smoothing filter. The joint bilateral filter using Gaussian has two major parameters: the spatial domain standard deviation σ_g and the range domain standard deviation σ_f. While σ_g steers the spatial size of the blur, σ_f steers the sensitivity to edges in the filtering process. A small σ_f increases the edge sensitivity and a large σ_f increases the smoothing effect. Likewise, a large σ_g increases the area of the blur, while a small σ_g does not reduce the noise.

IV. Experiments

In order to experiment with the models we used a 2D convolution layer for all the layers, The standard input size of 256 x 256 was used for input with activation function as ReLu, in some of the layers the stride size was increased. the size of the kernel was 3x3. in the case of Domain adaptation we were down-sampling the image hence we used an up-sampling layer also. In the case of Image re-colorization, the images were kept to be of the same size, and only convolutions were done on the data. We used 'adam' optimizer for the compilation. also, mean squared error was used to check for the error rate and loss. Other parameters were varied across all the experiments to get the results.

A. RGB to Temperature Mapping

We experimented with models consisting of 1 to 6 convolutional layers followed by 0 to 3 dense layers with layer sizes of 32, 64, 128 in all possible combinations. Each convolutional layer is followed by a ReLU activation layer and a Max pooling layer. Each dense layer is followed by a ReLU activation layer. The kernel size of the convolution kernels is the same for all convolutional layers. To compensate for the skewed data we weighted the dataset to reach balanced data.

We used 'sparse categorical cross-entropy as a loss function for all the models as 'categorical cross entropy' gave extremely poor results. The optimizer used was 'adam'. Some models were also used with every single R, G, B channel. However, no significant accuracy result was found so we went back to the complete RGB image.

1) Result for a dataset I: As given in the TABLE I, results for all the combinations with 1 2 3 dense layers, 1, 2, 3, 4, 5 convolutional layers of layer size 32, 64, 128 were near the accuracy range of 20% to 50% with losses in the range of 2 to 4. Among these, 3-conv-1-dense-64-nodes gave the best result of 52% accuracy with 2.64

Sl.	Model	Accuracy	Loss
1	1-conv-32-nodes-0-dense	19%	3.06
2	2-conv-32-nodes-0-dense	14%*	2.72
3	3-conv-32-nodes-0-dense	9%*	2.74
4	4-conv-32-nodes-0-dense	19%	2.74
5	5-conv-32-nodes-0-dense	14%	2.77
6	1-conv-64-nodes-0-dense	50%*	2.77
7	2-conv-64-nodes-0-dense	9%*	2.72
8	3-conv-64-nodes-0-dense	23%	2.86
9	4-conv-64-nodes-0-dense	19%	2.74
10	5-conv-64-nodes-0-dense	28%	2.73
11	1-conv-128-nodes-0-dense	19%	2.70
12	2-conv-128-nodes-0-dense	50%*	2.73
13	3-conv-128-nodes-0-dense	19%	2.72
14	4-conv-128-nodes-0-dense	19%	2.75
15	5-conv-128-nodes-0-dense	50%*	2.72
16	1-conv-32-nodes-1-dense	19%	2.77
17	2-conv-32-nodes-1-dense	19%	2.76
18	3-conv-32-nodes-1-dense	9%*	2.85
19	4-conv-32-nodes-1-dense	9%*	2.73
20	5-conv-32-nodes-1-dense	14%	2.75
21	**3-conv-64-nodes-1-dense**	**52%**	**2.64**
22	4-conv-64-nodes-1-dense	23%	2.74
23	5-conv-64-nodes-1-dense	23%	2.73
24	1-conv-128-nodes-1-dense	19%	2.73
25	2-conv-128-nodes-1-dense	19%	2.71
...	...	...	...

TABLE I: Results for the CNN models used on dataset I. *refers to the models which gave variable results on repeated tests

2) Result for IITJ dataset: We used IITJ dataset and re-ran the models with new data. Similar to the first dataset, accuracy and loss were calculated for each of the models of CNN with 1 to 6 convolutional layers followed by 0 to 3 dense layers with layer sizes of 32, 64, 128 in all possible combinations. The results can be found in TABLE II.

3) Result: Due to the very small number of videos present in the RGB dataset, the RGB image to temperature mapping gave results that were highly inaccurate and misleading. Some of the models gave results that were 100% accurate from the start and had zero loss throughout the training.

Sl.	Model	Accuracy	Loss
1	1-conv-32-nodes-0-dense	100%	0
2	2-conv-32-nodes-0-dense	100%	3.5380e-04
3	3-conv-32-nodes-0-dense	70%	1.50
4	4-conv-32-nodes-0-dense	83%	0.95
5	5-conv-32-nodes-0-dense	58%	1.73
6	1-conv-64-nodes-0-dense	100%	0
7	2-conv-64-nodes-0-dense	100%	0.05
8	3-conv-64-nodes-0-dense	64%	1.11
9	4-conv-64-nodes-0-dense	45%	1.49
10	5-conv-64-nodes-0-dense	38%	2.83
11	1-conv-128-nodes-0-dense	100%	0
12	2-conv-128-nodes-0-dense	19%*	3.20
13	3-conv-128-nodes-0-dense	58%	2.06
14	4-conv-128-nodes-0-dense	22%	2.15
15	5-conv-128-nodes-0-dense	54%	3.60
16	1-conv-32-nodes-1-dense	83%	4.16
17	2-conv-32-nodes-1-dense	96%	0.15
18	3-conv-32-nodes-1-dense	48%	3.27
19	4-conv-32-nodes-1-dense	29%*	2.59
20	5-conv-32-nodes-1-dense	16%*	2.32
21	1-conv-64-nodes-1-dense	100%	9.1443e-06
22	2-conv-64-nodes-1-dense	100%	6.8134e-04
23	3-conv-64-nodes-1-dense	80%	1.16
24	4-conv-64-nodes-1-dense	74%	2.08
25	5-conv-64-nodes-1-dense	41%*	2.13
...	...	...	...

TABLE II: Results for the CNN models used on dataset II. *refers to the models which gave variable results on repeated tests

B. RGB-NIR and NIR to Temperature Mapping

The processed data is fed through the convolutional layers. There are multiple convolutional layers with no pooling layers in Image re-colorization approach to prevent the loss of data and multiple pooling layers in Domain adaptation, in up-sampling and down-sampling. The activation function of the convolutional layers is the ReLU function. The kernel size of the convolution kernels is the same for all convolutional layers. More details for both datasets are given below.

Experiments with EPFL dataset : For this problem, we used auto-encoder neural networks to generate the NIR images from the RGB images. Our initial approach to this problem was to solve with domain adaptation technique where the image is first reduced to very little data and reconstructed to get a new image. This approach is useful in the problems where auto-encoders are used to generate new data from some sample of existing data. However, these models require a lot of training data and huge resources. Since we did not have a large dataset to train the model and due to the restraint on the computational power that we had access to, the images were not of good quality. Hence we dropped this approach as the images generated by these networks were extremely poor and no useful property could be extracted from it.

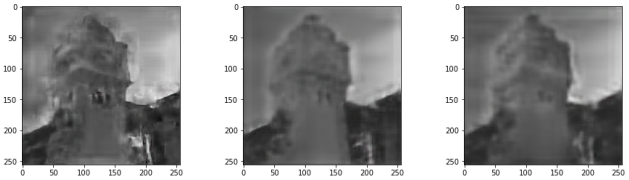
The Initial model used a convolutional layer of size 256-128-128-64 for the down-sampling followed by 128-128-256 for up-sampling. They were trained for 1000 or 100 epochs. Each convolutional layer in the down-sampling part had a max-pooling layer following it. Fig. 3 (a) is the final image generated by this model. Other models that gave better results were,

- 16-8-8 for 1000 epochs
- 64-128-128-256-256-256-128 in down-sampling and reverse in up-sampling for 1000 epochs
- 128-64-32-16 in down-sampling and reverse in up-sampling for 100 epochs
- 128-64-32-16 in down-sampling and reverse in up-sampling for 100 epochs
- 256-128-128 in down-sampling and reverse in up-sampling for 100 epochs
- 256-128-128 for 1000 epochs

And many other models of the combination 256, 128, 64, 32 for 100 epochs gave even poorer results. Note: Each input image was of the size 256 x 256. Some results with the domain adaptation approach where the pooling was done are:

(a) (b)

Fig. 2: (a) Input RGB image. (b) Ground Truth.



(a) (b) (c)

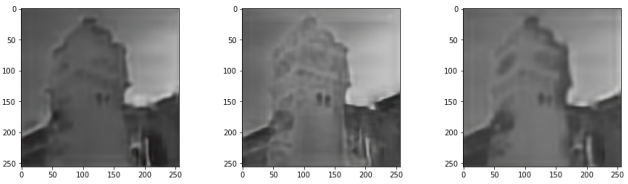
Fig. 3: Test Images generated by Domain adaptation approach (a) 256-128-128-64 for 1000 epochs, (b) 32-8-8 for 1000 epochs (c) 16-8-8 for 1000 epochs.

Thus we shifted to Image re-colorization approach models. Here we did not use the pooling layer to reduce the data size and instead kept the size the same throughout the model while varying the convolutional layer sizes. We used different combinations of layer sizes from 256 to 64 or 32 size convolutional layers. Some results with the Image colorization approach where the pooling was done are given in Fig 4.

Other prominent models which gave good results were,

- 256-256-128-128-64 for 1000 epochs
- 128 64 64 32 for 100 epoch
- 128 128 64 64 64 32 for 1000 epochs

Finally, the best results, shown in Fig 5, came with the model 128-128-128-64-64-64-64 for 1000 epochs size convolutional layers. For all our models we used ‘adam’ optimizer and mean squared error as loss function. This approach also had some issues with the shadow regions.



(a) (b) (c)

Fig. 4: Test Images generated by Image colorization approach (a) 64-128-128-256-256-256-128 for 100 epochs, (b) 512-256-256 for 100 epochs (c) 256-128-128 for 100 epoch

Fig. 5: Final image generated from the model 128-128-128-128-64-64-64 for 1000 epochs

After the image is generated by the convolutional network. The image is passed through a joint bilateral filter to remove the haze around the edges. The original output image shows high noise in the edges due to the sub-correlation property of the convolutional layer which causes the edges to lose sharpness. To compensate for this, the joint bilateral filter is used. We use a Gaussian distribution for the weights of the nearby regions.

(a) (b) (c)

Fig. 6: Images generated after post-processing (a) Ground Truth vs Images generated after (b) filtering and (c) contrast adjusting

Experiments with IITJ dataset : To Train the model the images were resized to 256 x 256 pixels. The NIR images consisted of some colored boxes and text due to the camera features so the NIR image was first adjusted to Grayscale and a 3 channel image was generated from it for training. Then we used various training models to get the NIR images. Some of the results are given in Fig 7.

Result : The EPFL dataset had the appropriate number of RGB and NIR image pairs. Each of the images had a varied intensity of light illuminating the whole area and various colors across the pixels. This enabled the models to learn how

(a) (b) (c)

Fig. 7: Test Images generated by Image colorization approach (a) 128-128-128-128-64-64-64 for 100 epochs, (b) 128-128-128-128-64-64-64 for 1000 epochs (c) 256-256-256-256-256-256-256 for 100 epochs

the intensity of each color and the individual brightness of each pixel will be translated from an RGB image to a NIR image. Therefore the models were able to predict the intensity of each pixel within an acceptable margin of error to give proper and acceptable NIR images.

However, the IITJ dataset had a fixed background with a lot of unwanted pixels and colors in the NIR image which made the models unable to learn the proper transformation of RGB to NIR for the skin. Moreover, the images present in this dataset were taken from a different perspective and had a lot of unwanted detail with low resolution causing the results to be inadequate with very high error.

V. Conclusion

The prediction of the temperature in the first part is accurate for more than 50% of the time, however, this accuracy may be misleading due to the small dataset that we have used. The results obtained by the convolutional network are very similar also due to the insufficient dataset present, Also which leads to very high loss function values. Similarly, IITJ dataset was very small for the proper prediction of the temperature and mostly consisted of 97 °F. Due to this, the training was bad and as a result, the accuracy was misleading. Therefore, this experiment should be repeated with a larger dataset.

There is a need for a varied dataset that must have multiple colors in the images and must have been shot in multiple brightness and conditions in order to get a proper dataset. In the case of the human body and skin, there must be a proper image of the skin without any kind of distortions as the range of temperature variation along various images of skin and the human body will be low so an even more efficiently captured dataset must be used.

References