Unilever Capstone Project

Automatic Photodamage assessment from Facial Image

Mingrui Liu, Shuyu Huang, Woo Jin Kim,
Yuheng Shi, Yunjun Xia

12/09/2020

1. Motivation – Why this project?

The **Objective** of this project is to build a model by leveraging deep learning and image analysis to obtain **automatic grading** on mottled hyperpigmentation(patches of skin become **darker** in color than the normal surrounding skin) conditions from facial images taken in clinical studies.

The need for automating the judgement arises because,

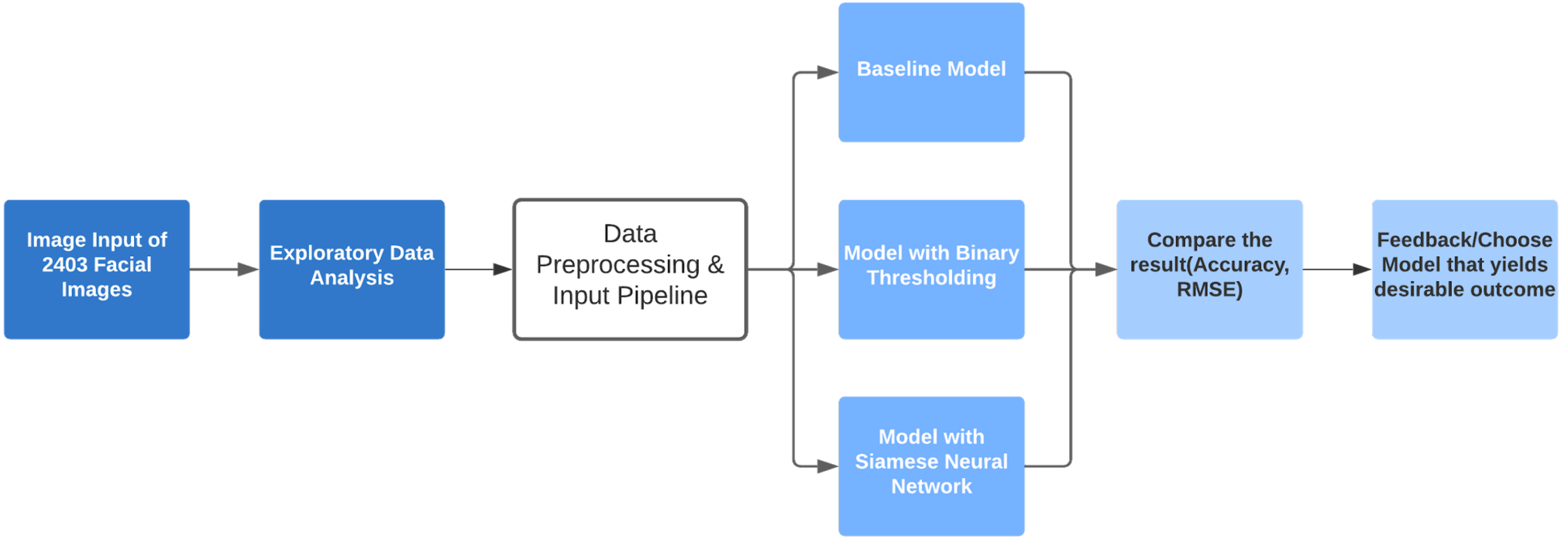
- Limited availability of expert graders
- Differences between graders
- Some subjectivity and variability of the experts

Previous attempt by Unilever to build multivariate models on image analysis parameters were **not successful**. Similarly, deep learning with VCG16 transfer learning **did not yield** the desired outcome.

With given 2403 image data, our **goal** is that other data science methods, such as a knowledge-based approach or combinations of techniques may have potential to result in desired outcome.

Auto Grading **Image Analysis** **Facial Image** **Deep Learning**

2.1 The Overview – Flowchart



2.2 The Overview – Models & Goal

As you can see in the flowchart, we tried **3 different models**. We need to achieve the goal of getting Root mean square error less than **0.3**.

Those 3 models are

- Baseline model – With ResNet50 transfer learning with GlobalAveragePooling and Dense Layer
- Model with Binary Threshold – Implemented Binary Thresholding on top of Baseline model
- Model with Siamese Neural Network – New model with Siamese Neural Network

Details of those models will be discussed in modeling section. We compared those 3 models' output with/without image pre-processing. We managed to settle with **Siamese Neural Network model** since it yields promising output which is very close to our goal of this project.

3. Dataset

We were given 3 different types of data.

- Jpeg **image** files (2403 files)
  - only skin is retained. Hair, eyebrows, eyelashes, nostrils, and mouth are **masked out** for **privacy issue**.
- Excel files for **grades**
- Pdf file for **grading criteria**

Privacy Issue:

- Unilever followed strict guidelines as informed by **law**, Unilever policies and subjects' informed consent.
- **pseudonymized** both in identity (filenames) and in content.

SUBJID	side	VISITNUM	VISIT	Mottled hyperpigmentation
1	Left	1	Baseline	2.5

3.1 Dataset – Grading Criteria

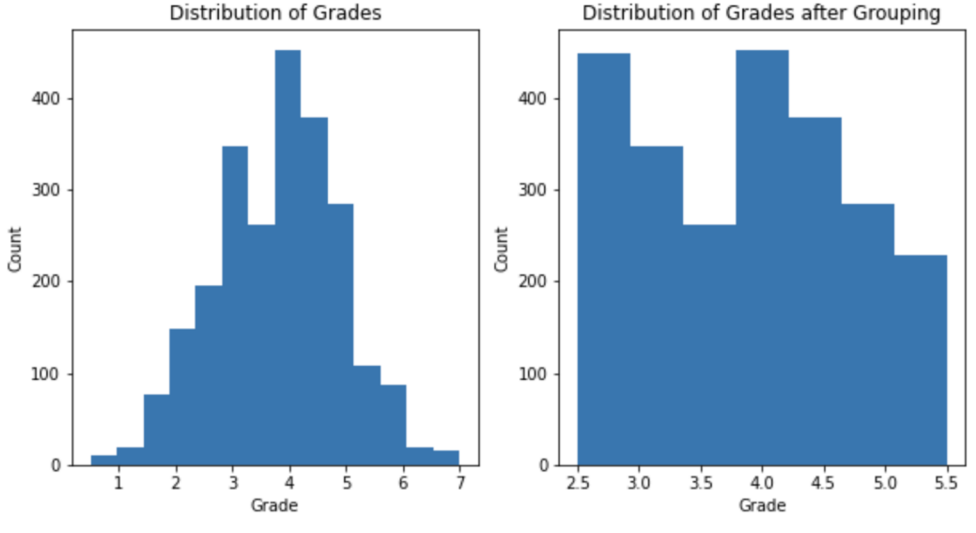
Each of our image files, contain corresponding **grade**. From 0 to 9 with interval of 0.5. The grades are separated by following category/condition.

Mottled Hyperpigmentation (Entire Side of Face)				
Grade				Description
0			None	No hyperpigmentation
1	2	3	Mild	None/very few small or faint brown patches not obvious from a distance
4	5	6	Moderate	Obvious presence of brown spots, which may be faint but covering the entire face, or very pronounced but more localized
7	8	9	Severe	Severe –moderately/severely pigmented brown patches often with whitish spots (mottled). Very noticeable from a distance

Note that we grouped 1,2,3 as mild, 4,5,6 as moderate and 7,8,9 as severe.

4.1 EDA – Grade

- **Imbalanced dataset**
  result in **low accuracy** when predicting facial photodamage condition with extreme patterns.
- **Solution: grouping**
  grouped all the images which have grade less than 3.0 as <3.0 group and grade greater than 5.0 as >5.0 group, it is fairly balanced.



distribution before grouping

distribution after grouping

Grade	Image count
0.5	10
1.0	19
1.5	76
2.0	148
2.5	196
3.0	348
3.5	261
4.0	452
4.5	379
5.0	285
5.5	108
6.0	87
6.5	18
7.0	16

image count before grouping

Grade	Image count
< 3.0	449
3.0	348
3.5	261
4.0	452
4.5	379
5.0	285
> 5.0	229

image count after grouping

4.2 Data Pre-Processing

Input Pipeline

- transformed all of our images into **tensor** with caching & prefetching.
  - significantly reduces the training time for each epoch by 70%.

Input Pipeline

Data Augmentation

We randomly **flip** images horizontally

- **Increase** the amount of the input data
- **Prevent** overfitting

Normalization

- **Converted** the image pixels to range from -1 to 1, as the original pixels range from 0 to 255.

  Resulting in **higher efficiency** in neural network

Order of Pre-Processing

Input Pipeline -> Data Augmentation -> Normalization

5.1 Modeling – Baseline Model

Baseline Model Construction

- Transfer Learning **ResNet50:** first 160 layers to be non-trainable
- **Dropout** layer with rate 0.5
- **Global Average Pooling** layer
- **Dense** layer as classifier with **softmax** activation function

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Functional)        (None, 16, 16, 2048)      23587712
_________________________________________________________________
dropout_1 (Dropout)          (None, 16, 16, 2048)      0
_________________________________________________________________
global_average_pooling2d_1 ( (None, 2048)              0
_________________________________________________________________
dense_1 (Dense)              (None, 7)                 14343
=================================================================
Total params: 23,602,055
Trainable params: 5,534,727
Non-trainable params: 18,067,328
```

5.1 Modeling – Baseline Model

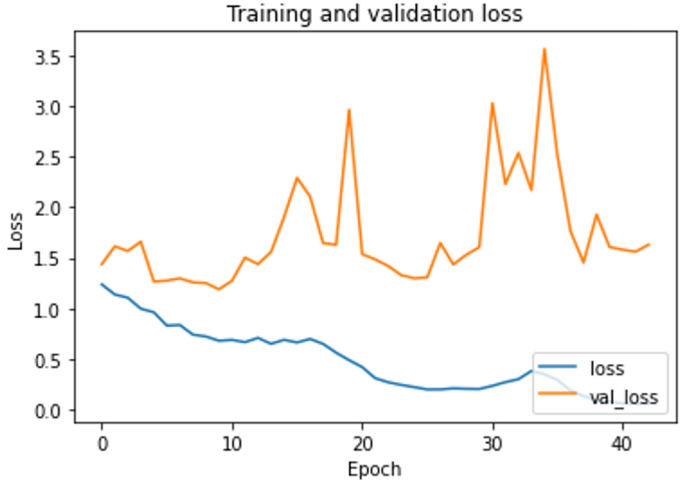
Baseline Model Results

- Optimizer: **Adam**
- Learning rate: **Exponential Cyclical Learning Rate**
- Loss function: **sparse categorical cross-entropy**
- Metrics: default **accuracy**
- Batch size: 32, Shuffle buffer size: 1024

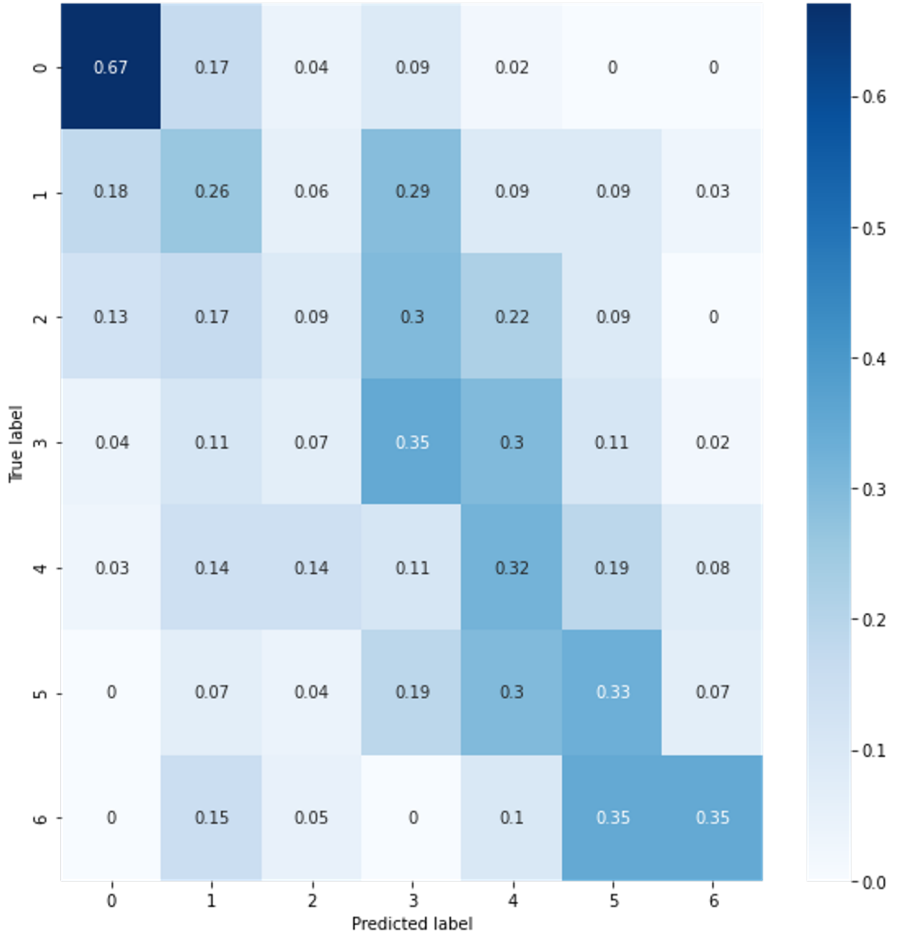
50 epochs
Training Accuracy: 99.94%
Validation Accuracy: 52%
Test Accuracy: 50.42%

```
[ ]  loss, accuracy = model.evaluate(test_ds)
     print("Accuracy: {:2.2%}".format(accuracy))

     19/19 [==============================] - 5s 256ms/step - loss: 2.4354 - accuracy: 0.5042
     Accuracy: 50.42%
```



5.1 Modeling – Baseline Model

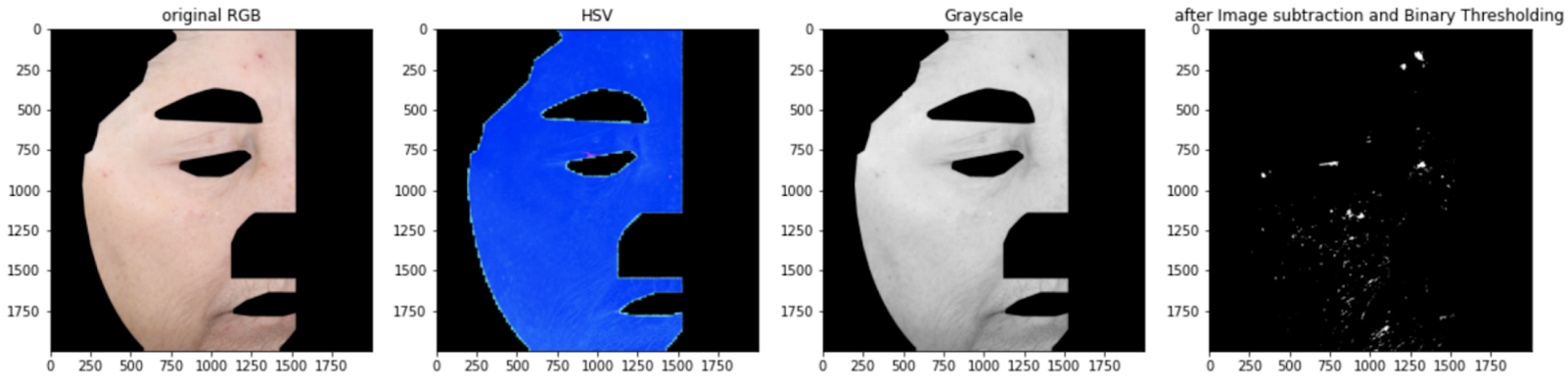


Confusion Matrix

5.2 Modeling – Binary Threshold

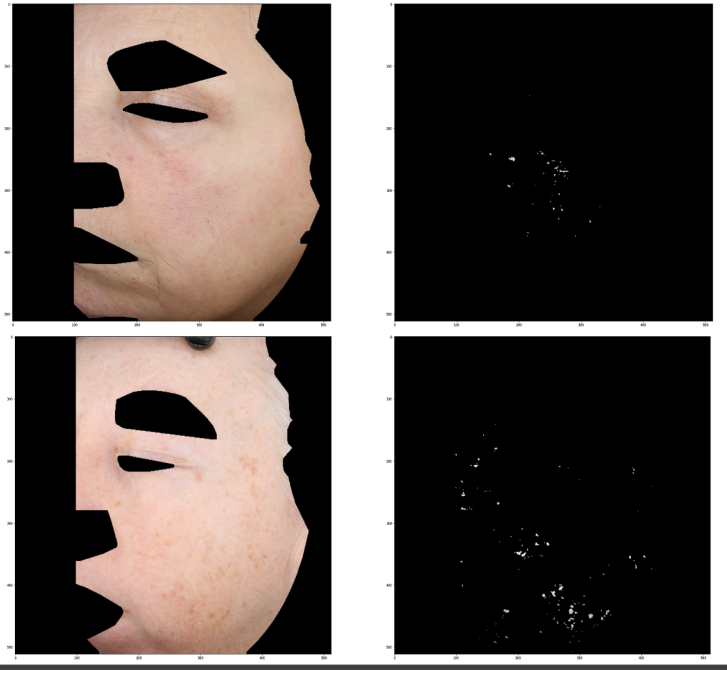
Our **Binary Threshold model** is the baseline model with Pre-processing of our image by extracting regions of interest which is mottled hyperpigmentation area on the face. Below, are the procedure of our modeling.

- **Image Subtraction**
  1) Convert RGB image to Grayscale color space to simplify the calculation
  2) Convert RGB image to HSV color space and extract brightness value(V) from HSV image
  3) Subtract Grayscale value out of normalized brightness value to reveal region of interest
  4) Multiply the binary image with the brightness layer of the HSV to get the brightness intensity binary threshold image



5.2 Modeling – Binary Threshold

- **Binary Thresholding to obtain binary image**
  1) Set the threshold value as 0.165 in this case for the best **visual result**. Any region above the value of 0.165 will be considered to be region of interest, and any region below will be discarded
  2) Below figures are the image of **grade 3(above) and 6(below)** before/after Image subtraction we mentioned in previous slide

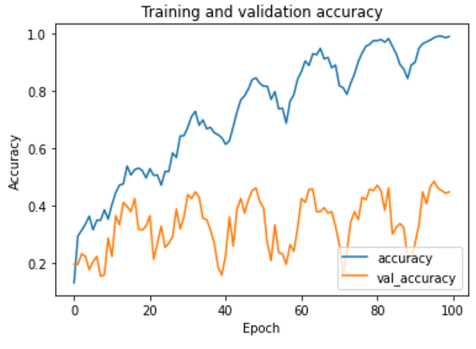


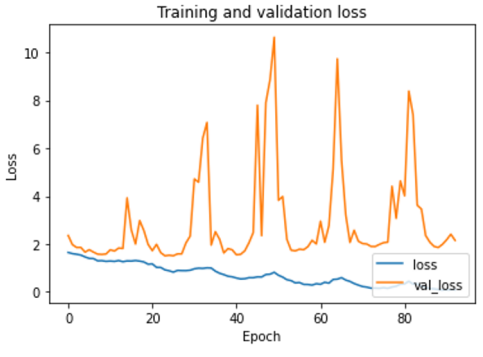
Both of the image yields similar output with the difference of **grade 6** image yields more white spot after binary threshold. Also we can see that white spot on **grade 6** image is **widely distributed**.

If you compare **original image** of both **grade 3 and 6**, you can visually tell there are more **brown**(mottled hyperpigmentation) spots on image of **grade 6**.

5.2 Modeling – Binary Threshold

After the Pre-Processing of the image using **Binary Threshold**, we used our Baseline model.






```
loss, accuracy = model.evaluate(test_ds)
print("Accuracy: {:2.2%}".format(accuracy))
```

```
8/8 [==============================] - 1s 146ms/step - loss: 2.3987 - accuracy: 0.4232
Accuracy: 42.32%
```

Highest accuracy we could obtain from this model was actually **45%** which does **not** show an **improvement**. It is even lower than our baseline model's validation accuracy.

Also, our test accuracy is **42.32%** which is lower than that of our baseline model.

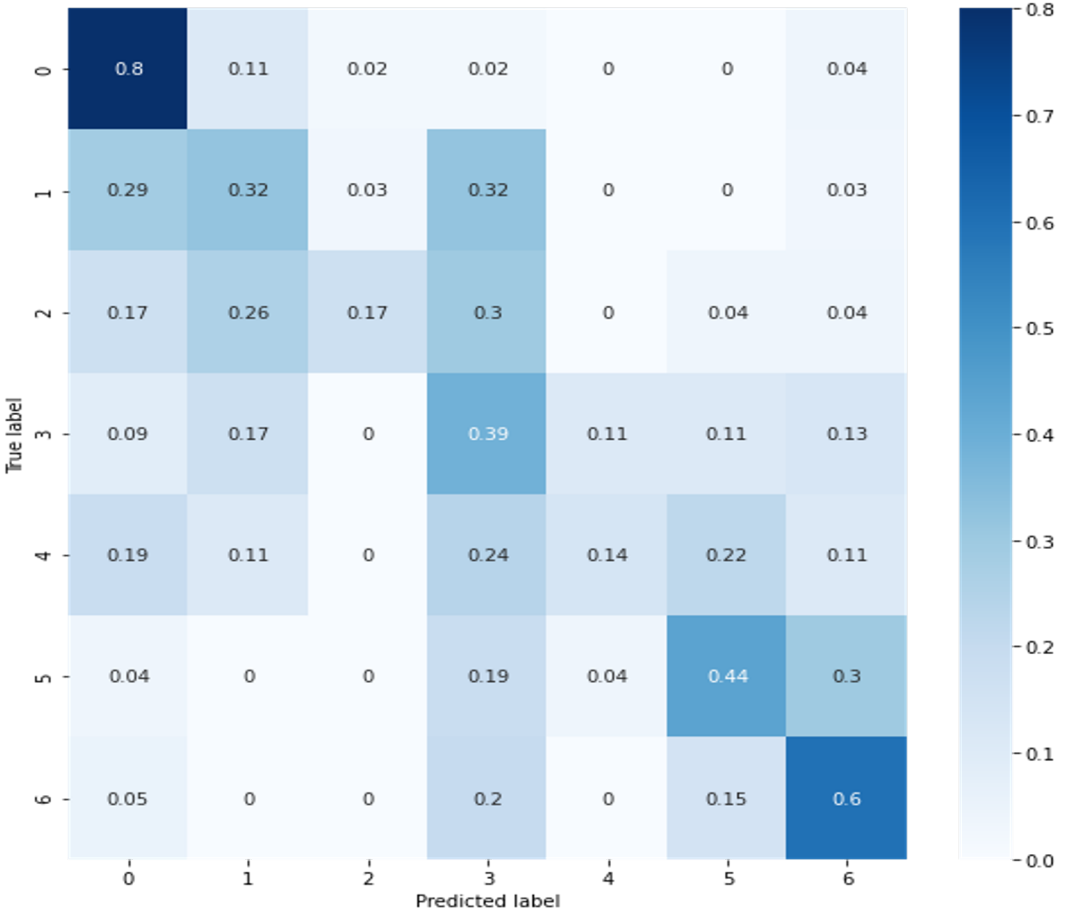
We suspect that the result we got is from the problem where our binary threshold cannot extract the all regions of interest or fail to exclude all irrelevant features.

For example, it might capture the shadow area around eye socket as region of interest when it is not.

5.2 Modeling – Binary Threshold

Figure on the right is **confusion matrix** of our Binary Threshold model.

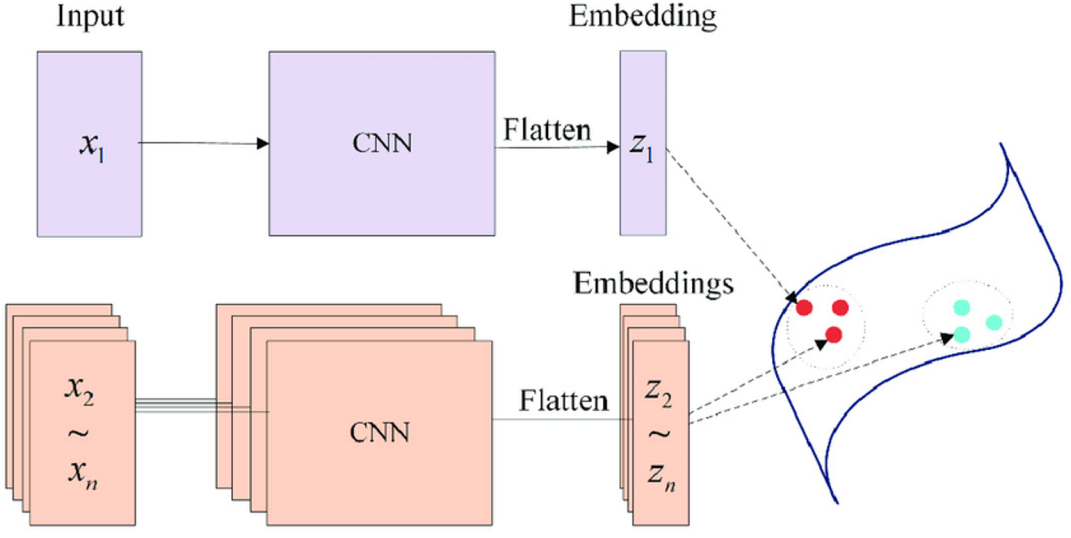
In addition to the interpretation we had to our baseline model, we can also notice that misclassification occurs near **grade 4**.



5.3 Modeling – Siamese Network

Siamese Neural Network is a class of neural network architectures that contains two or more **identical** sub-networks which expected to have **same configuration** with the **same parameters and weight**.

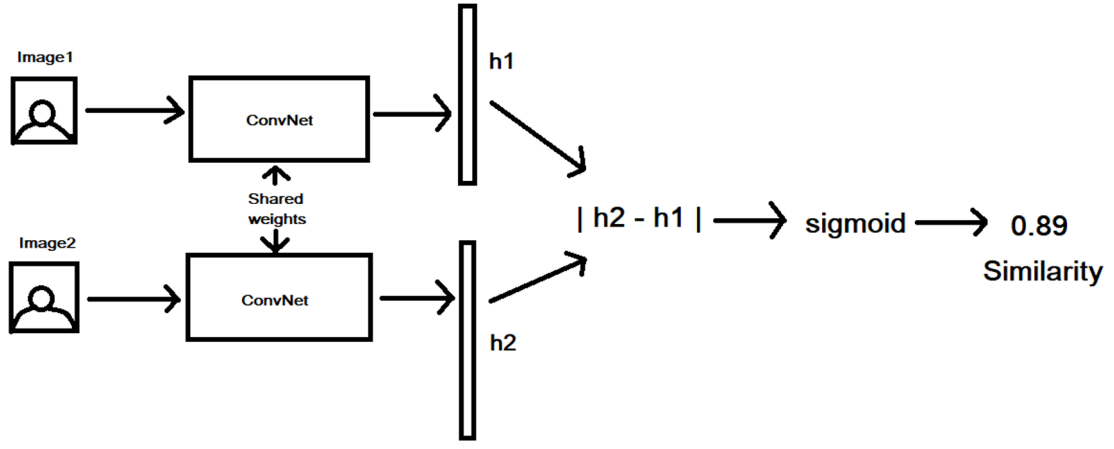
Siamese Neural Networks can find **similarity** of the input and all other dataset **by comparing** their feature vectors generated by the sub-network.



It properly solved the data lacking problem and generate better prediction.

5.3 Modeling – Siamese Network

This is the **algorithm** we used



Basic Algorithm

1) Two images are fed to separate CNN model.

2) The last layer of the CNN produces a fixed size vector .

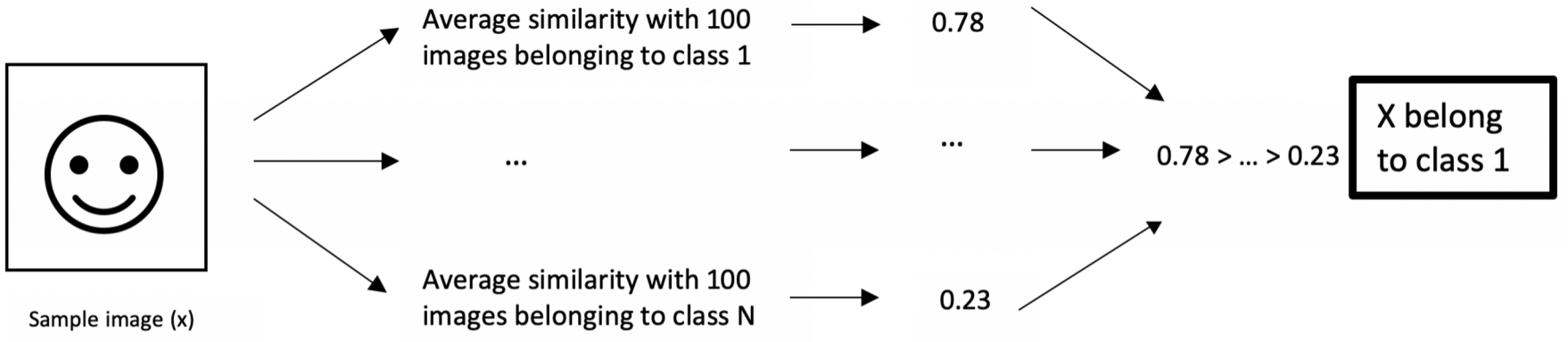
3) Calculate Euclidean distance.

4) The values then pass through a **sigmoid** function and a similarity score is produced.

5.3 Modeling – Siamese Network

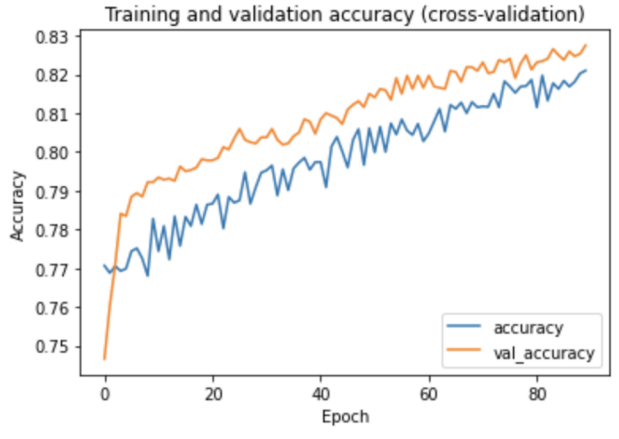
In order to perform actual classification on images, we follow,

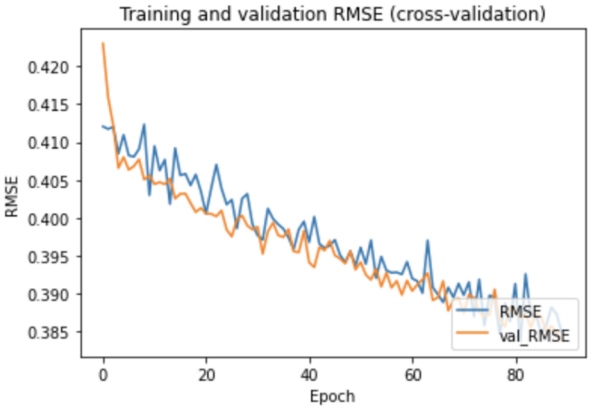
1. Group images from the data set based on their class
2. Calculate the similarities between the sample image and images for each class
3. Find the class which has the highest average similarity with the sample image
4. The class with the highest similarity will be the class prediction assigned to the sample image.



5.3 Modeling – Siamese Network

Using **Siamese Neural Network**, it yields the promising result.





5-fold cross-validation:

accuracy: 0.827

loss 0.479

RMSE 0.383

5.3 Modeling – Siamese Network

```
loss, accuracy, RMSE = siamese_network.evaluate(test_ds,steps= 32)
print("Loss: {:1.2}".format(loss))
print("RMSE: {:1.2}".format (RMSE))
print("Accuracy: {:2.2%}".format(accuracy))
```

```
32/32 [==============================] - 6s 195ms/step - loss: 0.4804 - accuracy: 0.8125 - root_mean_squared_error: 0.3900
Loss: 0.48
RMSE: 0.39
Accuracy: 81.25%
```

The test Root Mean Square Error is **0.39** and the accuracy is **81.25%** which shows the consistency with our validation set.

5.5 Modeling – Comparison Chart

	Baseline Model	Binary Threshold	Siamese Neural Network
Training Accuracy	99.94%	99.69%	99.99%
Validation Accuracy	52%	45%	84%
Test Accuracy	50.42%	42.32%	81.25%
Improved?	Baseline	No	Significantly

6.1 Next steps – Baseline Model

Use ResNet152 ImageNet Caffe

We used ResNet50 as our transfer learning model. Throughout our research, we found that other pre-trained models such as **ResNet152 ImageNet Caffe** may have better performance.

In the research paper **Assessing the Severity of Acne via Cell Phone Selfie Image Using A Deep Learning Model** by Hang Zhang, they used **ResNet152 ImageNet Caffe** and achieved Root Mean Square Error of **0.482** which is better than target performance bar **0.517** which was set by Nestle Skin health and Microsoft as the **highest** Root Mean Square Error among 11 dermatologists.

We will try to train this model and **compare** the result between ResNet50 in our Next step.

6.2 Next steps – Binary Threshold

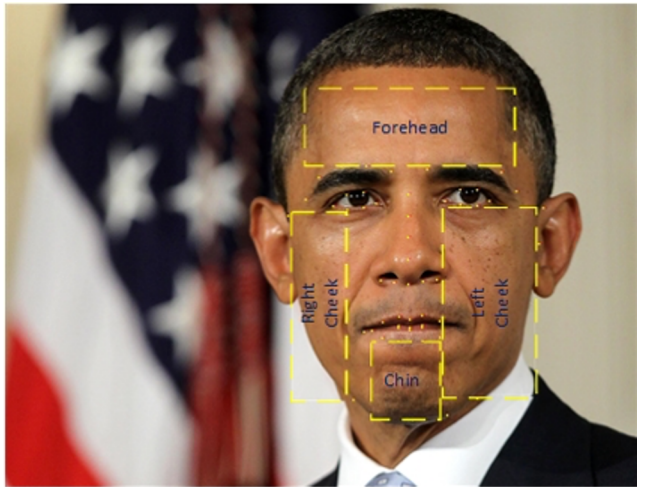
Extract Regions of Interest more precisely

As mentioned on Binary Threshold slides, we suspect our extraction of the Regions of Interest have some problems. It might be the case where our model extracts **rosacea** (a common facial skin condition that causes redness and visible blood vessels) as mottled hyperpigmentation. We can try to improve this using some extra color intensity procedures.

We also expect to try out to exclude **shadow area** around eye sockets. These area is usually darker than facial skin and they are easily extracted as Regions of Interest of mottled hyperpigmentation.

Crop our initial input image

We could try to extract skin patches from facial images. From the forehead, both cheeks and chin of each face image by applying the pre-trained **Shape Predictor 68 Face Landmarks (landmark model)** published by Akshayubhat on github.com. Then we will calculate the weighted average score of the whole face.



7. Conclusion

What now?

We tried lots of different Preprocessing method along with different types of model. In early stage of our project, we tried different types of models that were not introduced in this slides.

After few trial errors and editing, we managed to narrow it down to 3 specific models.

Baseline model (Convolutional Neural Network with ResNet50), **Binary Threshold** with Baseline model and **Siamese Neural Network Model**.

Our team chose **Siamese Model** as our final model candidate regarding the result **comparison** between three models introduced.

We would also like to push and explore little further with our **Binary Threshold** method as well.

Thank you for watching!

Appendix

1. T. Chantharaphaichi, B. Uyyanonvara, C. Sinthanayothin and A. Nishihara, "Automatic acne detection for medical treatment," *2015 6th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, Hua-Hin, 2015, pp. 1-6, doi: 10.1109/ICTEmSys.2015.7110813
http://www.cephsmilev2.com/chanjira/Papers/2015/ICICTES2015Thanapa.pdf

2. T.Chantharaphaichit, Thanapha, B. Uyyanonvara, C. Sinthanayothin and A. Nishihara. "AUTOMATIC ACNE DETECTION WITH FEATURED BAYESIAN CLASSIFIER FOR MEDICAL TREATMENT." (2015).
http://www2.siit.tu.ac.th/bunyarit/publications/2015_RIIT2015_BKK_Khim.pdf

3. Will Koehrsen, "Neural Network Embeddings Explained"
https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526

4. Hang Zhang, "Assessing the Severity of Acne via Cell Phone Selfie Images Using A Deep Learning Model"
https://devblogs.microsoft.com/cse/2019/02/05/assessing-the-severity-of-acne-via-cell-phone-selfie-images-using-a-deep-learning-model/

5. Sean benhur J, "A friendly introduction to Siamese Networks"
https://towardsdatascience.com/a-friendly-introduction-to-siamese-networks-85ab17522942