# A REPORT

# ON

# FACIAL DEMOGRAPHICS

BY


| | |
|---|---|
| Shreyansh Joshi | 2018A7PS0097G |
| Ashutosh Sharma | 2018B3A70928P |
| Vikas Sheoran | 2018B3A70847H |
| Jash Shah | 2018A8PS0507P |


AT

**Silver Touch Technologies, Ahmedabad**

A Practice School-I station of

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**June, 2020**

A REPORT

ON

# FACIAL DEMOGRAPHICS

BY


| | | |
|---|---|---|
| Shreyansh Joshi | 2018A7PS0097G | Computer Science |
| Ashutosh Sharma | 2018B3A70928P | Computer Science & Economics |
| Vikas Sheoran | 2018B3A70847H | Computer Science & Economics |
| Jash Shah | 2018A8PS0507P | Electronics & Instrumentation |


Prepared in partial fulfillment of the
Practice School-1 Course No.
BITS F221

AT

**Silver Touch Technologies, Ahmedabad**

A Practice School-I station of

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**June, 2020**

# Acknowledgement

The PS-1 project has been a great opportunity to work on real world problems and research for optimal and professional solutions.

Firstly, I express my deepest gratitude to Dr.S.Gurunarayanan Sir, PSD Head for managing this wonderful initiative of Practice School and  for providing us with this great opportunity to learn how the corporate world works.

I would like to thank Mr. Adarsh Parikh, Associate Vice President and Training and Development Head at Silvertouch Technologies, for providing us with an excellent opportunity to intern with them and gain exposure to work and Professional ethics.

It is my radiant sentiment to place on record my best regards and deepest sense of gratitude to Mrs/Ms. Tanisha Bhayani, AI/ML engineer at Silvertouch and our Project Manager and Mrs./Ms. Akansha Goplani, HR at Silvertouch and our project  coordinator, for their mindful guidance which was extremely valuable for the project, both theoretically and practically.

I would also use this opportunity to extend my deepest gratitude and special thanks to our PS faculty, Dr. Aruna Malapati , Associate Professor(CSIS), BITS-Pilani, Hyderabad Campus, who, despite being extraordinarily busy with her/his duties, took time to hear each issue, guide and keep me on the correct path throughout this time.

Lastly, I would like to extend my thanks to my project team members who have been highly supportive. The project is a collective effort of all the members, and it would not have been possible without them.

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI (RAJASTHAN)**

**Practice School Division**

**Station:** Silver Touch Technologies Ltd **Centre :** Ahmedabad

**Duration:** 6 Weeks **Date of Start:** 18th May

**Date of Submission :** 23rd June

**Title of the Project:** Facial Demographics

**ID No./Name(s)/ Discipline(s)/of the student(s):**

| | | |
|---|---|---|
| Shreyansh Joshi | 2018A7PS0097G | Computer Science |
| Ashutosh Sharma | 2018B3A70928P | Computer Science & Economics |
| Vikas Sheoran | 2018B3A70847H | Computer Science & Economics |
| Jash Shah | 2018A8PS0507P | Electronics & Instrumentation |

**Name(s) and designation(s) of the expert(s):**

Tanisha Bhayani, AI ML Engineer, SilverTouch Technologies Ltd
Akansha Goplani , HR at SilverTouch Technologies Ltd

**Name(s) of the PS Faculty:**

Dr Aruna Malapati, Associate Professor(CSIS), BITS-Pilani, Hyderabad Campus

**Key Words :** Facial demographics, SVMs, CNNs, feature extraction

**Project Areas :** Machine Learning, Deep Learning and their applications in daily life

**Abstract:** The project Facial Demographics aims to take Computer Vision to a whole new level by creating intelligent machines that are able to predict the gender and age of a person, just by looking at his/her face image. This report throws light on the progress so far, showcasing various techniques used to complete the project.

Shreyansh Ashutosh Vikas

Signature(s) of Student(s) Signature of PS Faculty

Date: 23rd June 2020 Date:

# Table of Contents

# List of Illustrations

# 1. Introduction

Silver Touch Technologies is a leading and globally accepted IT Solution Provider and currently at the forefront of Digital Transformation & Emerging Technologies to serve the customers across the world.

Silver Touch Technologies Limited was founded in February 1995 with a view to deliver clients with the Information Technology Solutions in the fields of system integration and software Services. Now it is into multiple areas like System integration and Software Development. The Company is providing all end-to-end Information and Communication Technology (ICT) solutions to its clients under a single umbrella. It has carried out several national and international Joint Ventures which have enabled the company to expand its geographical reach and create a diverse portfolio of clients. The company has executed various projects in India and other countries including USA, UK, France, Australia, Middle East and Germany.

As a part of the IT services provided by SilverTouch, we have been given an AI - ML project. The project is titled ‘Facial Demographics’ . We have to design a model for facial analysis which can estimate age and gender of a given image.

The objectives of the project are given below:

- To create a model to predict gender from a face image.
- To create a model to classify the face image in one of the classes of the age.
- To create a model to estimate exact integer age from the face image.
- Create a script (and a service) that takes the input as an image and gives the age class, estimated age and gender as output with its relevant scores.

The report is divided into five sections. First section defines the problem statement in detail and Second section  gives a gist of literature reviewed for the project.The third section explains in detail the methodology used and is further divided into four sub sections namely EDA, CNN, TL and ML. Fourth section talks about the future work and Fifth section concludes the report respectively.

## 2. Problem Definition

One of the hottest fields today in Computer Science, Human-Computer Interaction and HMI fields, has the problem of designing an intelligent machine that can estimate the age and gender of a person on its own, using face images, using well known concepts in artificial intelligence. Many big companies around the globe, such as IBM, Google  today, are investing in this field. The primary reasons for the same are -

- The boom in image data in recent times
- The boom in computational power in recent times, allowing for better training of deep learning models
- The fact that we can make a machine decipher certain characteristics of humans just by looking from its face, is in itself very intriguing.

The ability of a machine to figure out a person's gender and age (or even race for that matter), could prove to be a very useful aspect in real life. Since  age and gender are 2 crucial factors in identifying a person by face, the model could be deployed in real life to find lost persons. So, the model could take as input race, age and gender of the lost person (given by someone who wants to find him/her) and the machine would then scan  whichever face it gets, will check if the predicted age, gender and race match with the given information. That way AI powered devices could help find a missing person in real life.

This technology could also be applied to improve the vision of the humanoid robots and automated systems

This is precisely the problem the project “Facial Demographics” deals with. It aims to analyze images, their distribution and return information on age, gender for each input face image, based on its facial characteristics. The created model, first extracts features (based on facial characteristics) and then uses those extracted high-dimensional features, to make predictions.  Some of these features could be wrinkles, hair color, presence of facial hair, etc.

This report contains 3 types of models, that were created for the same purpose, namely -

- CNN
- Transfer Learning
- Classical Machine Learning model

Each model takes as input an image of fixed size (varying from model to model) and outputs the required class/value.

## 3. Literature Review

We began our work by reviewing literature available on the problem. The research papers provided us valuable insight in the domain of ‘Facial Demographics’. Here are the research papers that we went through:

*Table I. List of Research Papers*

| Research Paper | Author(s) |
|---|---|
| Gender Classification Techniques: A Review | Preeti Rai and Pritee Khanna |
| Face Recognition Performance: Role of Demographic Information | Brendan F. Klare, Mark J. Burge, Joshua C. Klontz, Richard W. Vorder Bruegge, Anik K. Jain |
| Face Recognition and Age Estimation implications of Changes in Facial Features: A Critical Review Study | Rasha R. Atallah, Amirrudin Kamsin, Maizatul A. Ismail, Sherin A. Abdelrahman, Saber Zerdoumi |
| Age estimation via face images: A Survey | Raphael Angulu |
| Convolutional Neural Networks for Age and Gender Classification | Ari Ekmekji |
| Efficient facial representations for age, gender and identity recognition in organizing photo albums using multi-output ConvNet | Andrey V. Savchenko |

These  papers talked about various strategies such as different ways of feature extraction such as Gabor filters, LBP, etc that can be used with classical ML models. The papers also showcased the various CNN architectures used to solve the problem. They compared trainable and non-trainable algorithms' performance wrt. to the demographics of individuals and how the composition of the demographic information in the dataset influenced the performance. The papers, finally pointed out the importance of facial demographics for many applications such as facial recognition, age estimation etc. in numerous areas such as security, law enforcement, biometrics, forensics etc. and challenges involved such as pose, illumination, expression, aging, ethnicity, lifestyle, environment, databases,dataset.

# 4. Dataset Used

Initially, the dataset we chose for training our model is the **UTKFace** dataset (aligned and cropped). It is a large-scale face dataset consisting of 20,708 face images. All images are 200x200 with annotations of age (0-116), gender (Male, Female), and ethnicity (White, Black, Asian, Indian, and Others). The images cover large variation in pose, facial expression, illumination, occlusion, resolution, etc. We chose this dataset because it was a relatively smaller dataset and was appropriate for initial testing purposes. The images being already cropped and aligned saved us from doing that in preprocessing.

Later on after our models were completed and a baseline accuracy was reached, our mentor told us to try using another bigger dataset also. For this we used the **Wikipedia** Dataset. It consists of 62308 images crawled from all profile images from pages of people from Wikipedia with the same meta information. In the Wikipedia dataset, the age labels were assigned by first removing the images without timestamp (the date when the photo was taken). Then, assuming that the images with single faces are likely to show the personality and that the timestamp and date of birth are correct, a biological (real) age was assigned to each such image.

Note: An interesting observation, unlike the UTK Face dataset, the Wikipedia dataset is prone to have sampling bias.As most of the actors/celebrities/rich/famous personalities (though not always) appear much younger than someone the same age who is not a celebrity. These celebrity images are likely not representative of the general population and may cause a bias for making predictions on faces of the normal population.

# 5. Methodologies

This section of the report, describes in detail various techniques and models that have been created in the project. That is followed by the results obtained by different models.

**Note:** All the code was written in Python and Keras Framework was used as an API which is in turn built on top of Tensorflow.

## 5.1 Exploratory Data Analysis (EDA)

EDA refers to the critical process of analyzing data sets so as to discover patterns, spot anomalies, understand the data distribution , summarize their main characteristics with the help of summary statistics and graphical/visual methods. It was done to understand the data first and try to gather as many insights from it. EDA helped us in making sense of data in hand, before we got our hands dirty with it. Some of the EDA techniques we used on the metadata, and the interpretations are as follows:

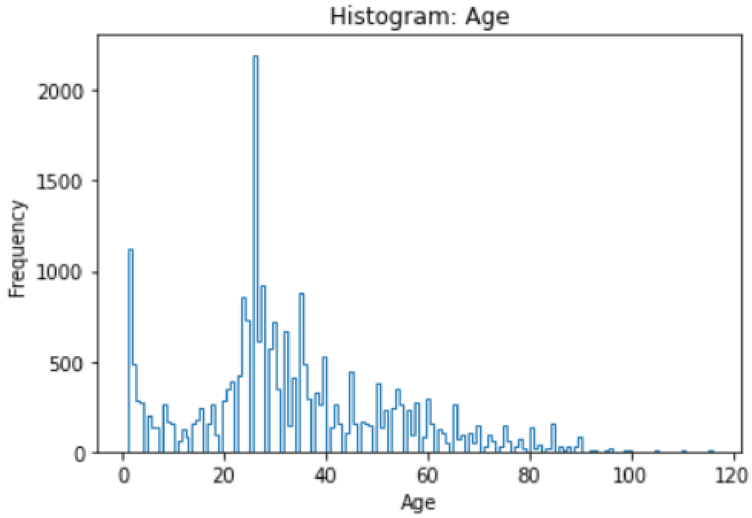



*Fig 1 : Histogram of the dataset ( Frequency vs Age)*

Fig 1. shows that the dataset contains relatively more images with ages between 25-30 and age 1 whereas there are too few images with ages > 80 and ages between 5-15. This to say, tells us that the dataset unbalanced. Now, if we use all images as it is for training, there are much more chances that the ages 25-30 may overfit. So, this tells us that we need to do some preprocessing before using the data to train. We have two options, either we remove the images from ages which have more or we increase the images for ages with lesser (by data augmenting).

```
Male_Age.describe()              Female_Age.describe()

count     12391.000000           count     11314.000000
mean         35.695666           mean         30.678186
std          19.705223           std          19.752001
min           1.000000           min           1.000000
25%          25.000000           25%          21.000000
50%          34.000000           50%          26.000000
75%          50.000000           75%          37.000000
max         110.000000           max         116.000000
Name: Age, dtype: float64        Name: Age, dtype: float64
```

*Fig 2. Descriptive Statistics of the dataset for the male and female*

Fig 2. shows the stats of the dataset, just at a glance we can know the number of images in the dataset gender wise, the mean, variance and the quartiles of the ages gender wise. This shows us the distribution of our data and this information comes handy during preprocessing of images.

```
# Dropping the rows with error values
train_new = df_train.dropna()
test_new = df_test.dropna()
```

*Fig 3. Finding unlabelled, corrupt images and removing them*

Having unlabelled images, corrupt images can cause problems in training the model, so we have to get rid of such images. Fig 3. shows this method of cleaning the data. We have made sure that there are no unlabeled images (NaN values essentially).

Some of the other techniques we used were cross-labels mean, variance, histograms, PMFs, IQR, Pie charts, Scatter plots, Box plots, Bee swarm plots, Violin plots, Count plots, etc. After the EDA on the metadata of the dataset was done, we went on to do the EDA on the actual image pixel data. Some of the techniques and the interpretations are as follows:

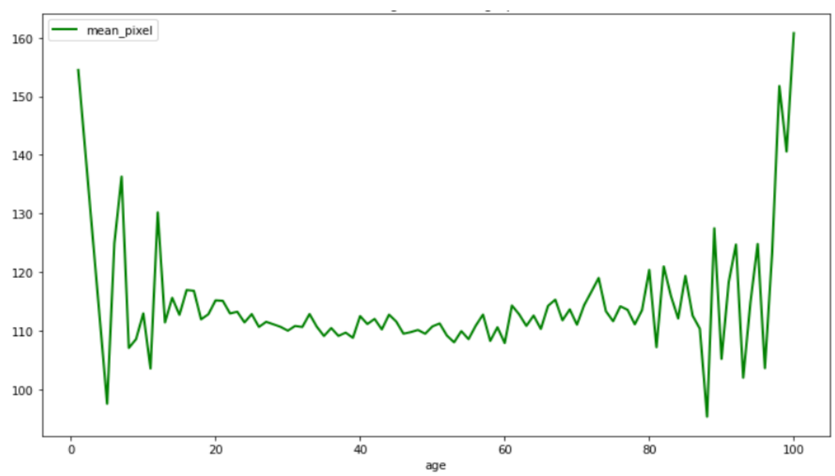



*Fig 4. Relation between age and average pixel values*

Fig 4. shows the graph plotted between age and average pixel value of the images of the WIKI dataset. Interestingly enough, we found that the average pixel value of middle ages (20-80) seems to be varying over a small range whereas average pixel values of both higher and lower ages seems to be varying over a much bigger range . This is because the majority of images have ages between 20 and 80. On either end of those, there exist outliers.

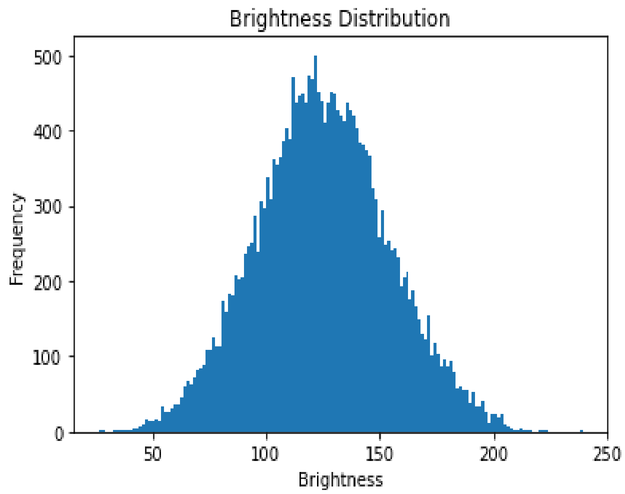



*Fig 5 : Distribution for Brightness of the images*

When distribution for brightness of all the images of the UTK Face dataset was plotted as shown in Fig 5, It came out to be an almost normal distribution with Mean 125.145 and Standard Deviation 28.991.

The following function was made for calculating the brightness:

```
def brightness(im):
    stat = ImageStat.Stat(im)
    r,g,b = stat.mean
    return math.sqrt(0.241*(r**2) + 0.691*(g**2) + 0.068*(b**2))
```

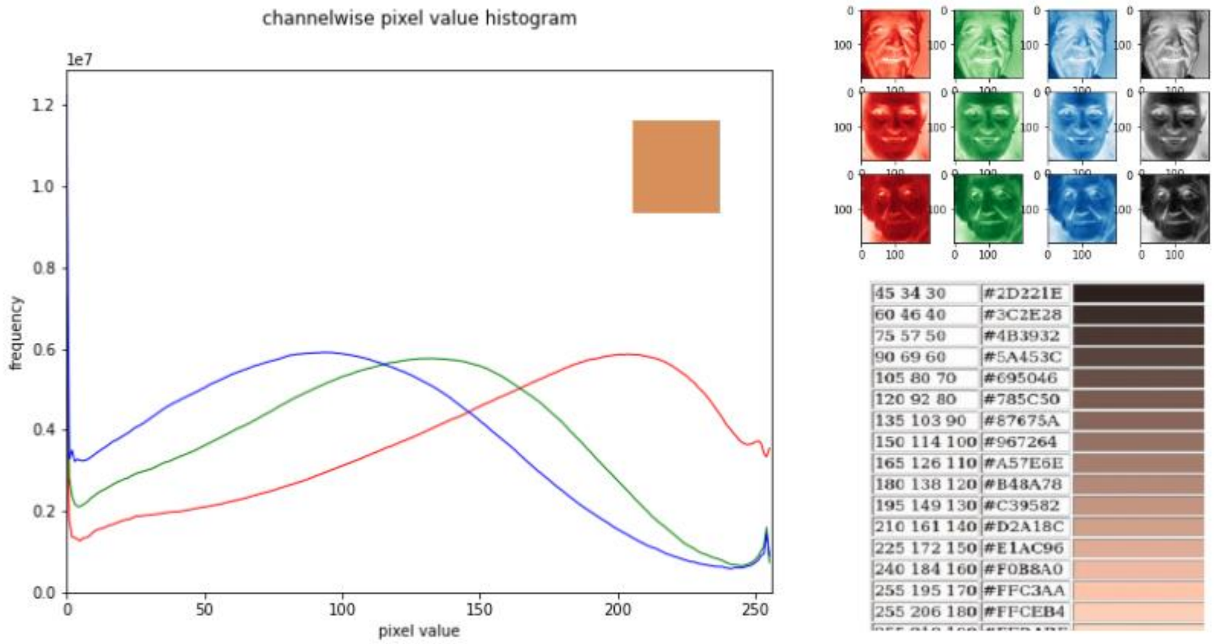



*Fig 6 : Channel wise pixel value histogram*

Fig 6. shows the histogram plotted for channelwise pixel values of R,G and B of images of UTK Face dataset.This also seems to have a symmetric distribution with R > G > B for skin tone on average

From the plot it's also visible that zero pixel value has some freq, which indicates that grayscale images could be present. On checking, we found 611 grayscale images.

## 5.2 Convolutional Neural Networks (CNNs)

Images are everywhere around us these days. In the field of DL, when it comes to dealing with images, the most easy to use, comprehensive and ubiquitous way is using CNNs.

Typically, CNNs have 2 main types of layers - **Conv layers**, that are the real feature extractors and **Max Pool** layers, that help in reducing dimensionality , i.e downsampling of features that have been extracted by Conv layers before.

## Models -

### 1. Gender and Age Classification -

The architecture that was used for gender/age classification tasks is described below -

- Basic idea was to have a block of convolutional and pooling layers, that help the model learn features and representations, followed by a set of FC layers, which are used to classify.

- My model took an input image of size (198,198,3). It was a RGB image.

- To predict both age and gender at once, I used a multi-output classification technique. This involved having a common part of feature extraction for both age and gender, but separate FC layers. The branching point is just after features have been extracted for both. I kept the feature extraction part common for both age and gender classification, because as per researchers, features are more or less along the same lines for predicting age and gender.

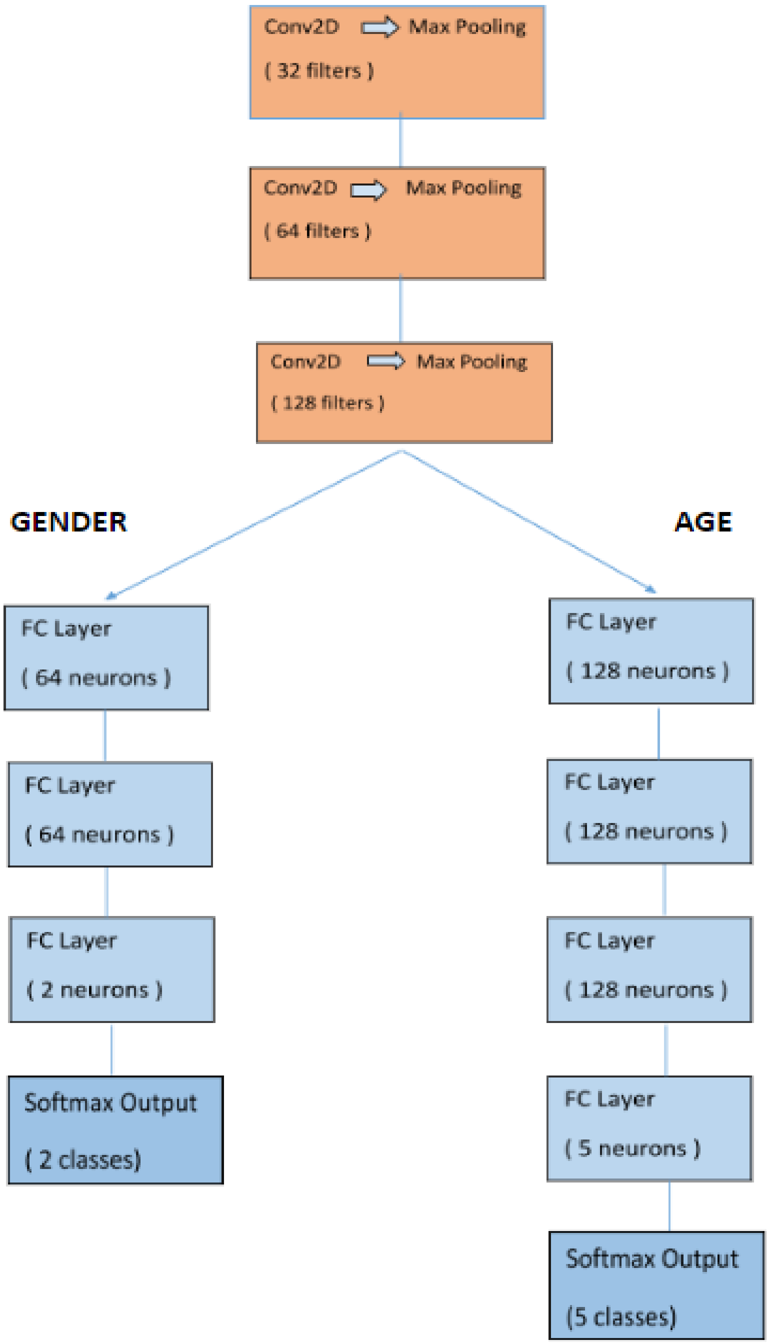



*Fig 7. CNN Architecture for age & gender classification*

- In both these architectures, I used L2 regularization in every FC layer and

Dropout (0.3) after every 2 FC layers.

- I used ‘Learning Rate Scheduler’ of Keras. Initial LR= 0.004 and it halved every 25 epochs.
- Loss functions and metric for both age and gender classification were “categorical cross-entropy” and “accuracy” , respectively.

- BatchNormalization was used after all layers (Conv &  FC) in both age and gender classification to normalize the output coming from them, for faster training. It also reduces what is called “covariate shift”.

## 2. Age Regression

The architecture that was used for gender/age classification tasks is described below -

- Basic idea was to have a block of convolutional and pooling layers, that help the model learn features and representations, followed by a set of FC layers, which are used to classify.
- Input to the model was a 180 x 180 x 3 image (RGB image).
- It had SeperableConv2D blocks, followed by MaxPooling2D and SpatialDropout2D intermittently.
- BatchNormalization was used after all layers (Conv &  FC)
- Loss function , this time was “mse” or mean-squared error.

- S.Conv2D -> (S.Conv2D) x 2 -> (S.Conv2D) x 2 -> Flatten() -> Dense1 -> (Dense2) x 2 -> Dense 3<br>64, 3x3 128, 3x3 256, 3x3 128 neurons 64 neurons 1 neuron

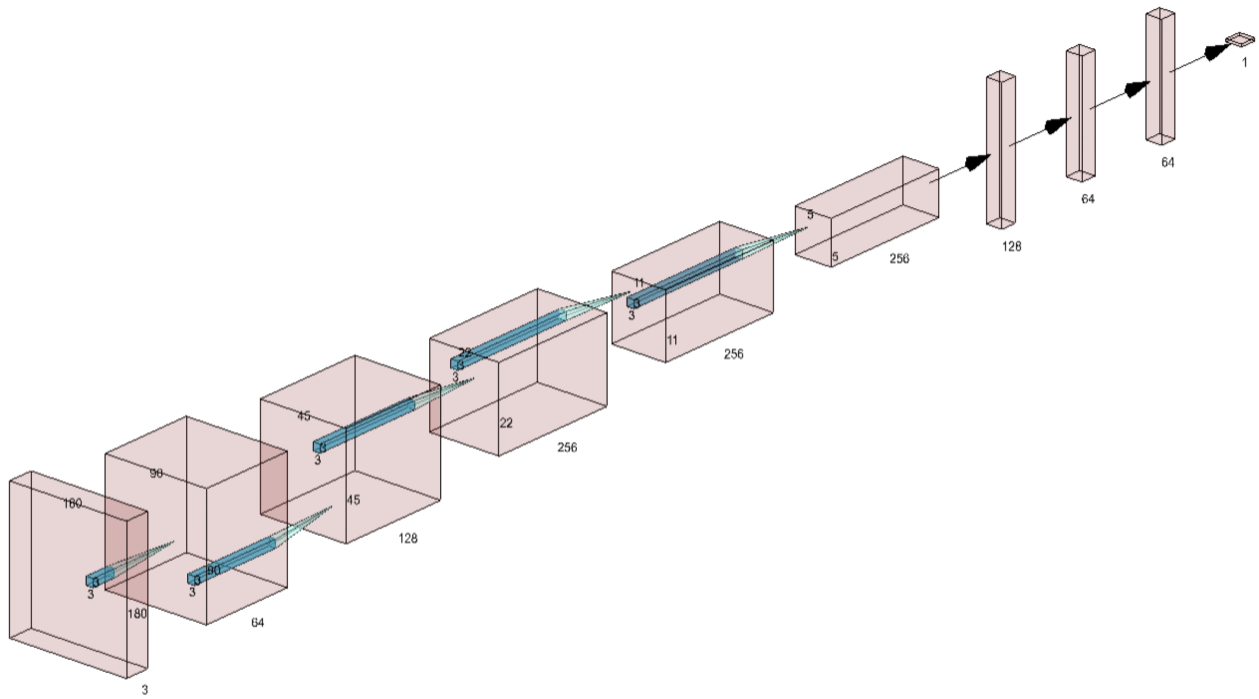



*Fig 8. CNN Architecture for age estimation (showcasing shapes for each layer)*

## Results Obtained:

### 1. Age and Gender classification -

Our CNN model, made from scratch was able to achieve a classification accuracy of 83.4 % on the test set for Age and 95 % on the test set for Gender.

For training sets, accuracy was about 96-97 % for age and 99 % for gender.

### 2. Age Estimation -

Achieved a validation mae of 5.46 and train mae of 5.20 & validation mse of 51.22 and train mse of 46.25.

Clearly, the model is not overfitting very much and good to make predictions !

## 5.3 Transfer Learning

One of the challenges we face in computer vision is the dimensions of feature input. A low resolution picture say, a 200 x 200 x 3 in our case will have 1,20,000 features for each data point and 1,20,001 parameters to train even for a Logistic Classifier which will give us limited performance. If we try a deeper fully connected system we could even have billions of parameters to train! Which will demand a huge amount of data and high computational power to provide decent performance. So we look for another solution and Convolutional Neural Networks(CNN) comes into the picture which attempts to look up for patterns in the image. In the first few layers of CNNs the network using convolutions can identify basic features like the edges and corners, and we can then pass these patterns down through our network and start recognizing more complex features as we go deeper. This property makes CNNs really good at identifying objects in images.

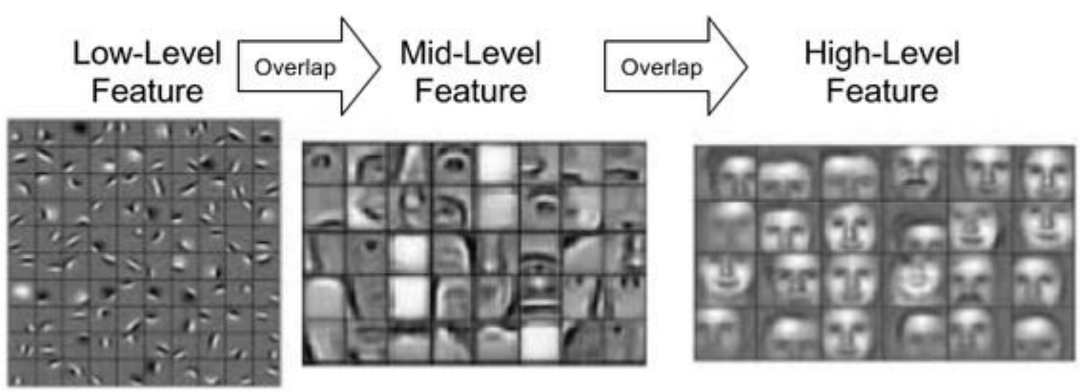
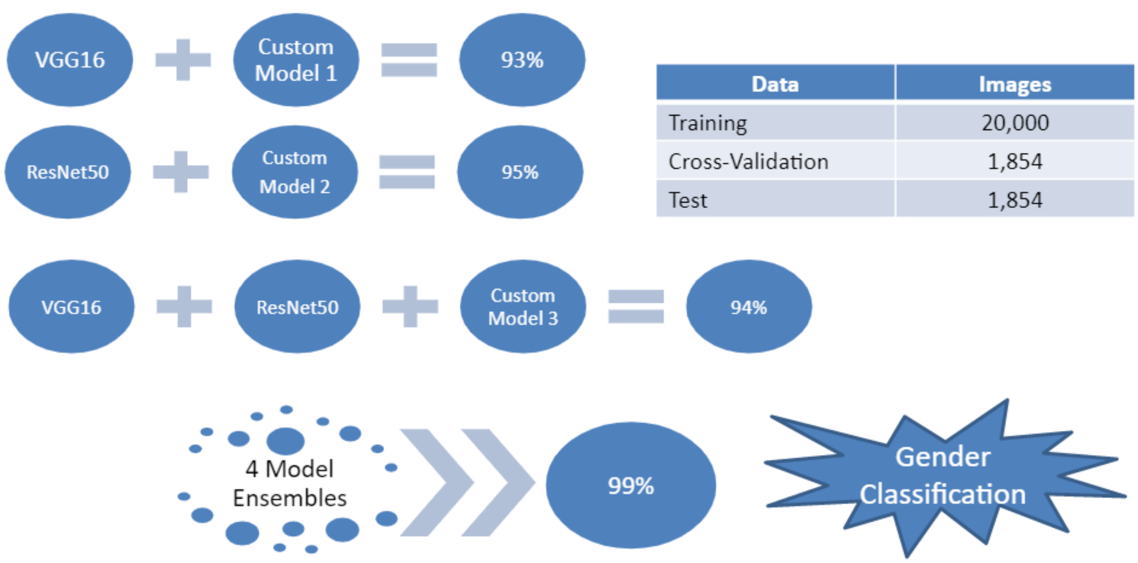



*Fig 9 : Feature Map in Convolutional Neural Networks (CNN)*

We leverage this property in Transfer Learning. We can utilize earlier layers of pre-trained CNN models to detect low-level features or even some mid-level features and save millions of parameters and boost performance by training a model on top of those matured features on limited data.

We utilized the **feature extraction** layers of Oxford VGGFace which is based on **VGG16** architecture by freezing the weights of the ConvNet part. We did a forward pass of the complete dataset to extract features and trained a custom fully connected model on top of it.

We repeated the same for **ResNet50** architecture and trained a different custom model on top of it. Then, we extracted features both from VGG16 and ResNet50 and did **feature fusion** and trained a model on top of that. The accuracy and train, test and cross validation split size for hold-out cross validation are as follow:




| Data | Images |
|---|---|
| Training | 20,000 |
| Cross-Validation | 1,854 |
| Test | 1,854 |

*Fig 10: Transfer Learning Architecture*

A significant class im-balance wasn't observed so classification accuracy is used as an optimizing metric though we obtained similar results for f1-score.

### Age Estimation

We also trained a transfer learning model for Age prediction as a real number (regression) on UTKFace dataset. The features were extracted from VGG16 which has a dimensional output of 6x6x512 for each 20,000 training, 1854 hold-out cross-validation and 1854 test data points. Custom models were trained on top of those extracted features, the specifics of which involves various combinations of Separable Convolutional layers, Max Pooling, followed by fully connected layers. The regularization techniques involve use of Spatial dropout which 'drops' feature maps of convolutional layers, Batch Normalization, Gaussian Noises, Dropout, and weight-constraints. The activation function used was the Exponential Linear Unit ( ELU ) in hidden layers, the activation function of the output layer was ReLU. The unbiased mean absolute error of each model trained is listed below:

*Table II. Regression of Age Model highlights, Results and Benchmarks*

| Model Highlights | Results | Benchmarks |
|---|---|---|
| • SeparableConv2D<br>• SpatialDropout<br>• BatchNorms<br>• Gaussian Noises<br>• Weight Constraints<br>• Dropout<br>• ELU | Ensemble MAE: 4.620075553157043<br><br>Model 1: 4.8693623457061745<br>Model 2: 4.885024872169711<br>Model 3: 4.862204356220563<br>Model 4: 4.8670852977201475<br>Model 5: 4.860798894619208<br>Model 6: 4.712113978633736<br>Model 7: 4.677876185933385<br>Model 8: 4.839006362061724<br>Model 9: 5.073820527636934 | Various Sources (UTKFace dataset)<br><br>• 9.19 [9]<br>• 7.36 [12]<br>• 5.44 [11]<br>• 5.39 [10] |

## Error Analysis

After getting the unbiased estimator of performance of the model a through error analysis was carried out using matplotlib, pandas and seaborn.

| | Age | Prediction | Mean Prediction | Error | Absolute Error |
|---|---|---|---|---|---|
| **1531** | 26 | 75.0 | 74.572762 | -48.572762 | 48.572762 |
| **1022** | 66 | 29.0 | 29.104818 | 36.895182 | 36.895182 |
| **1556** | 36 | 65.0 | 64.796783 | -28.796783 | 28.796783 |
| **587** | 9 | 37.0 | 37.025475 | -28.025475 | 28.025475 |
| **505** | 54 | 26.0 | 26.015465 | 27.984535 | 27.984535 |
| **...** | ... | ... | ... | ... | ... |
| **1442** | 26 | 26.0 | 26.025957 | -0.025957 | 0.025957 |
| **602** | 26 | 26.0 | 25.984459 | 0.015541 | 0.015541 |
| **1765** | 26 | 26.0 | 25.990316 | 0.009684 | 0.009684 |
| **400** | 28 | 28.0 | 27.991171 | 0.008829 | 0.008829 |
| **8** | 18 | 18.0 | 17.998865 | 0.001135 | 0.001135 |

1853 rows × 5 columns

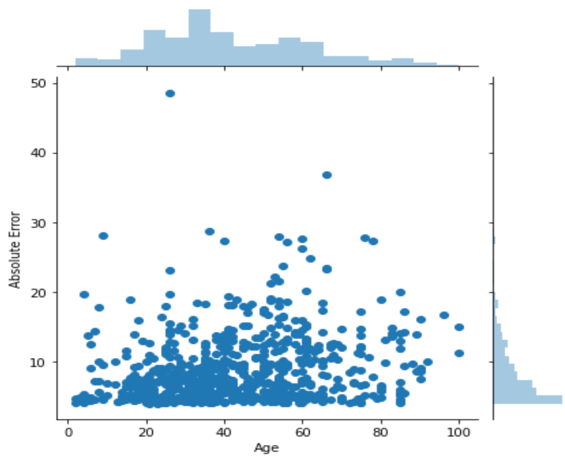
*Fig 11. Error Analysis Dataframe*

Since we have the indexes of images we manually explored the best and worst predictions on our test set. Model predictions were :

*Table III. Bad and Good predictions Analysis of Age Model*

| Bad | Good |
|---|---|
| • Low Resolution / Unreal<br>• Blacks<br>• Very Old<br>• Wrong Labels | • High Resolution<br>• Others<br>• 50 or younger |

Observing best 100 and worst 100 predictions. Further we dive deep into the error analysis to observe error variation with age.



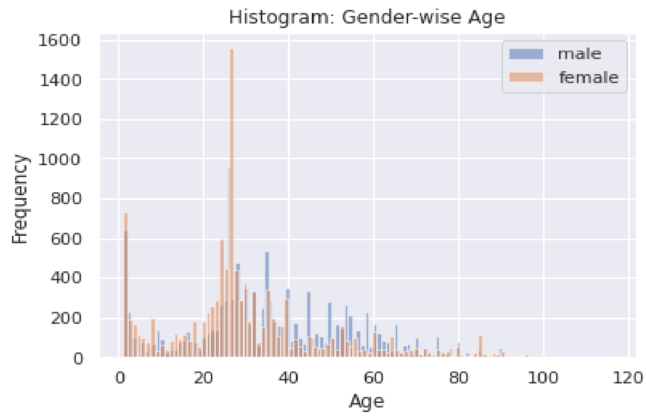



*Fig 12 : Age vs Absolute Error Analysis and Distribution of Age in the dataset*

As we can see from the above plot on the left that high absolute error was observed for the people with age greater than 70. Which is aligned with the insights we got from EDA on the distribution of the dataset and explainable. Moderate correlation of over 0.5 was found between age and absolute error.

## Deep Learning vs Core Machine Learning

A comparative analysis of core machine learning algorithms was made with our deep learning model for Gender Classification on features extracted from ResNet50 aka Transfer Learning using Core Machine Learning Models. The features extracted from ResNet50 have a dimension of 1x1x2048 which made it feasible for our analysis compared to 6x6x512 obtained using VGG16. The results obtained from *untuned* machine learning models are tabulated below:

*Table IV. Core Machine learning performance on Gender Classification*

| Model | Test |
|---|---|
| Neural Networks | 0.9395 |
| SVC | 0.9363538296 |
| LDA | 0.9363538296 |
| Voting Classifier - Top3 | 0.9336569579 |
| QDA | 0.9314994606 |
| Stochastic GBoost | 0.9309600863 |
| XGBoost | 0.9282632147 |
| Bagging | 0.9234088457 |
| AdaBoost | 0.9234088457 |
| Gradient Boosted Trees | 0.9207119741 |
| Random Forest | 0.9196332255 |
| LogisticRegression | 0.9190938511 |
| Perceptron | 0.9190938511 |
| Voting Classifier | 0.9174757282 |
| SGDClassifier - Linear SVM | 0.9099244876 |
| KNeighborsClassifier | 0.9088457389 |
| SGDClassifier - Logreg | 0.9034519957 |
| Linear SVC | 0.8948220065 |
| DecisionTreeClassifier | 0.8759439051 |

We strongly believe the relative performance of algorithms can vary highly when tuned and may even pass the deep learning approach. Furthermore, if we had time we would have utilized it tuning these machine learning models using Grid Search or Random Search followed by coarse to fine search.

### Multimodal Ensemble fusion

As mentioned earlier we used model ensembles to gain extra performance boost where we mean out all the probabilities for Gender Classification and Age for regression obtained from different transfer learning models. Then we questioned ourselves why restricted to mean?

So, we further took those transfer learning models' predictions standardized and normalized them using 'tanh' normalization and compared their respective performance by training a linear regression model and a support vector regression model. As expected performance was better than individual models but couldn't surpass simple mean combinations. This could be accounted to overfitting to the training ensembles even after trying regularization techniques. Again, if time permitted we would further work on this area finding the right ensemble fusion model.

## 5.4 Classical ML Techniques

Classical ML models are primitive ways of training a model and involve an extra step of manual feature extraction. This, some time, makes it a better and reliable option while working with small data. Also, at times, a better feature engineering algorithm can increase the overall accuracy of the model. Thus, Classical ML models were studied. SVM was chosen due to its relative high stability in bias-variance trade-off.

Support vector machines (SVMs) are formulated to solve a classical two class pattern recognition problem. We adapt SVM to face recognition by modifying the interpretation of the output of an SVM classifier and devising a representation of facial images that is concordant with a two - class problem. In the case of the Age Regression model, Support Vector Regression( SVR ) was used. An advanced model was also considered which initially consisted of a Multi-class classification and the regression on that dataset. This hybrid model gave much promising results, but took a lot of time  and hence is not suitable for real life applications.

After completing the General EDA, the entire successive process was divided into four parts : Face capture and recognition , Pixel Level EDA and correction, Feature extraction model and ML model deployment.

1) <u>Face capture technique</u> : This step involved the  frontal face capture technique. Two different methods were examined : Viola Jones method and HOG ( Histogram of oriented Gradients ). The former method involved using Haar like feature extraction followed by integral image computation and finally applying an adaboost cascade to it. This gave promising results when 2 or more faces are seen in the same image window. The HOG method, on the other hand, could only be applied to a single image and that too, it could not detect a very flat image. Hence, it needed high resolution images for working. After this step was completed, the facial image was cropped and augmented by computing the angle between the eyes. Two separate datasets were created, one with the grayscale image and other with the coloured image. This was done to both -  training dataset and testing datasets.
2) <u>Pixel Level EDA and correction</u> - This is one of the most important steps when working with Classical ML techniques. The 2 datasets were applied with different techniques. Initially, all the images were applied with Gaussian Blur so as to denoise images with high noise values. PNSR was

analysed to determine which images needed blurring. The grey scaled images were then applied with colour fixing techniques like Adaptive histogram equalization. The parameters were chosen by analysing the mean dark value of an image . It is quite important to note that the blurring stage was done before Contrast fixing. The details are given in the Appendix . On the other hand, the coloured images were convolved with Gamma Intensity Correction techniques. The value of Gamma was determined by plotting the mean brightness values of the images in the dataset.

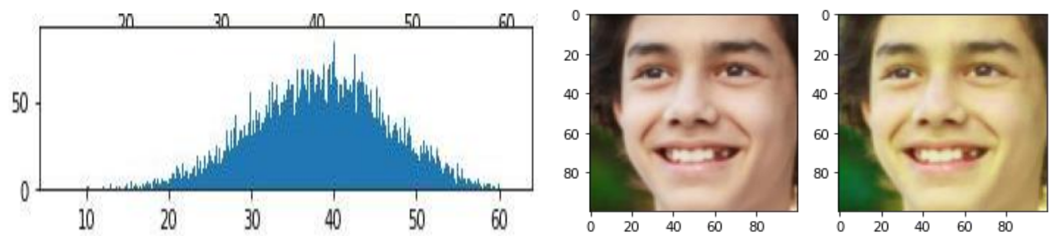


*Fig 13 : Mean brightness plot and the result of Gamma correction*

(3 different Gamma values were chosen according to brightness values : less than 30, 30 to 50, and greater than 50)

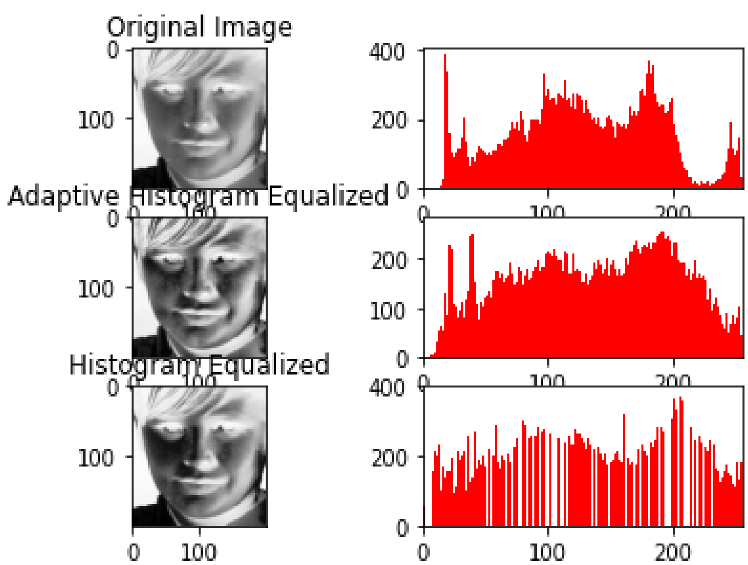



*Fig 14 : Adaptive Histogram Equalization*

Feature Extraction : This phase can be briefly divided into two phases : model free and model based algorithms:

**The model free algorithms** involved edge detection. This involved filters like Sobel, Laplacian and Log Gaussian convolutions. The filters could detect edges when applied in different orientations. An extra layer of Gradient thresholding was also applied. For the Grey scaled image, the filters were directly applied using Canny edge detection frameworks. For the coloured images, the filters were applied on the three channels separately and then thresholded using adaptive Gaussian thresholds.

**The model based algorithm** exploits the colour and texture information. The wrinkles and facial texture were extracted using Gabor filters followed by LBP (Linear binary patterns). The LBP was not applied directly because it needed to be customized for each image separately. In my method, an average LBP parameter was enough, because half the work was done by Gabor filters. The wrinkle, eyes and mouth were detected by this step.

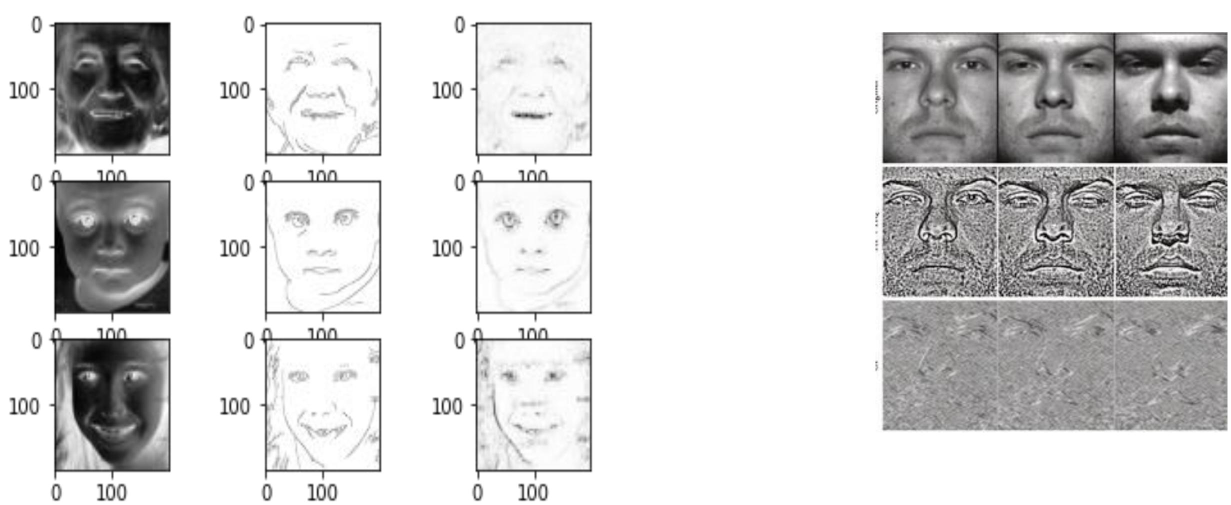



*Fig 15 : Sobel and Gabor filters on three different channels*

3) ML Model : This step firstly, involved the dimension reduction step with LDA and PCA for classification and regression respectively. The variance ratio was kept as 0.95( for PCA) . Different models were trained with permuting the various methods discussed above. Further, Adaboost SVM and Random forest - SVM were also examined. The several methods were ensambled for Greyscale and coloured images separately. The result obtained for the 2 datasets were further stacked and linear regression was applied. The SVM model itself contained a Weighted Voting technique ensambler for the four kernels used. The hyperparameter tuning was done using Grid Search with 5 fold CV. Random search and Bayesian Optimizers were very slow and incompatible with SVM and thus, were not examined.

   The SVM was accelerated in case of Age Regression model by the use of specially designed Hybrid model : The model first involved the classification of the age groups and then seperate age regression models were trained. The classification part involved the weighted voting output, thus, the probability of the second highest class was also considered. This improved the accuracy to an exceptional level. The figure below shows a brief of the method used.

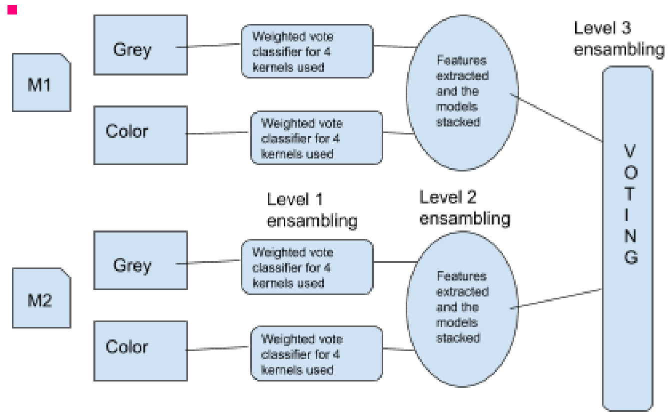



*Fig 16 : SVM model map*

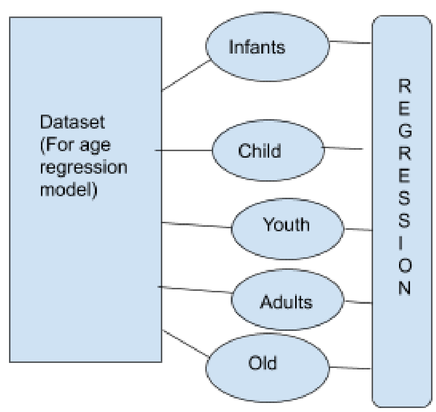



*Fig 17 : Hybrid model for age*

## Results Obtained :

The model provided an accuracy of about 94% on the Gender Classification. In case of age prediction, the model gave a range ( of 5 years ) of age with about 96.7% accuracy. The Age regression hybrid model gave about 2.46 MAE value!

## 6. Future Works

For the coming weeks we plan to work on improving the model and extending its capabilities :

1. Tuning the model for deployment in Real Life scenarios.
2. Creating Firmware based applications.
3. Model benchmarking and Research Paper based on the Analysis.

## 7. Conclusion

The project involves extensive analysis and classification of features. The initial Literature review helped us a lot in understanding the methodology and concepts involved in the project. Again, it is important to reiterate that the quality of the model is highly dependent on the quality of the data . Hence, we performed the EDA process in which we not only analysed local and global features, but also learnt about the data set. This helped us in determining the class weights.

Later on different models were created and benchmarked for their accuracy. We even created predictive models where the user can input an image of his/her choice. Interestingly, we observed the models are robust to covariate-shift and make impressive predictions.

# Appendices

## Appendix A - List of short forms used

*Table V: List of abbreviations used.*

| **Short Form (in text)** | **Full Form** |
|---|---|
| EDA | Elementary Data Analysis |
| CNN | Convolutional Neural networks |
| ML/ DL | Machine Learning/ Deep learning |
| MAE | Mean absolute error |
| SVM(C) | Support Vector Machines (Classifiers) |
| HMI | Human Machine Interaction |
| LDA | Linear Discriminant Analysis |
| LBP | Linear Binary Patterns |
| PCA | Principal Component Analysis |
| S.Conv2D | Separable 2D Convolution |

## Appendix B - Classical ML vs DL models

- Pros of DL
    - Best in class performance.
    - High accuracy
    - Scales effectively with data
    - No need for feature engineering
    - Transferable and adaptable

- Pros of ML
    - Work better and faster on small dataset
    - Financially cheap
    - Computationally stable
    - Easier to interpret.

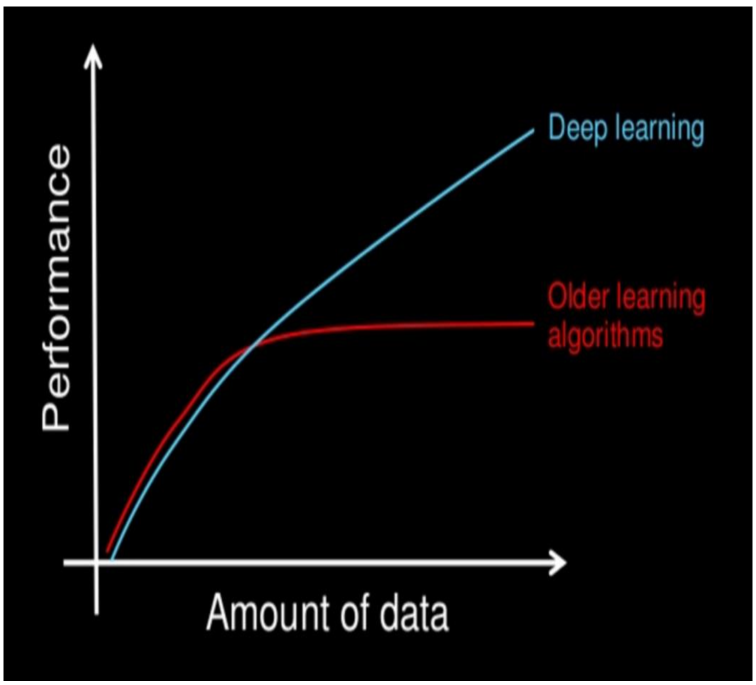



*Fig 18 : Performance vs Amount of data*

## Appendix C - Noise - Contrast Trade off in an image

An image needs to be pre-processed by improving its contrasts. There are many techniques for doing so, but the most famous ones are histogram equalization techniques. The technique works plotting the histogram of pixel values vs frequency of the image. The histogram is then stretched using various algorithms like normalization and hist-stretching, etc. This corrects the intensity of the images which have low contrast. For example :-

*Fig 19 : The left one is the original image and the right image is the corrected image.*

**The question now arises is that, is this method always useful?**

In the above image, it gave positive results. But, there are cases where this will fail. Consider a noisy image. This method will definitely increase the noise amount. One more major problem persists in case of thresholded images. This kind of images will be forcefully thresholded to have all the pixel intensities. Consider the following example :

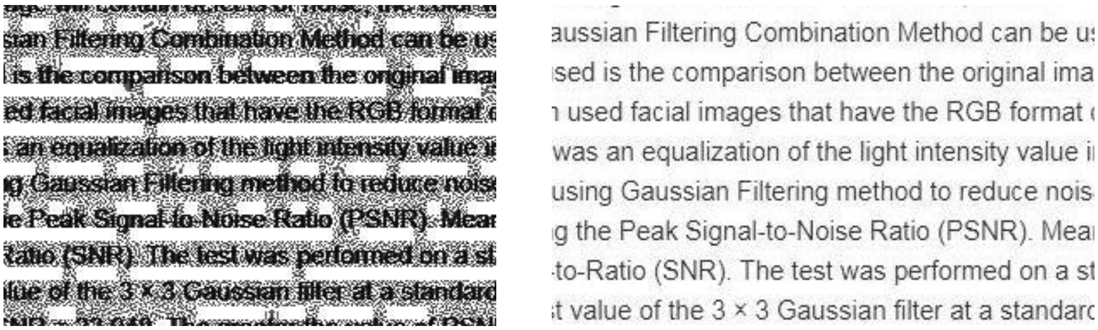



*Fig 20 : Thresholded image with its equalized output*

*Fig 21: Noisy image with its equalized output*

As we can clearly see, the method creates really bad output in both the cases. In such cases, the image first needs to be operated with PSNR to calculate the noise ratio. The ratio will help in solving the problem in 2 ways. It will allow us to know if the image needs to be denoised and, if the image is thresholded.

There are three possibilities :

1) PSNR value is very high - In such cases, the image first needs to be blurred. And then an adaptive histogram technique should be applied. This adaptive technique will intelligently equalize the histogram.
2) PSNR value is normal - Even here the adaptive technique will work better than other methods.
3) Low PSNR values - This indicates that the image is thresholded. So, histogram techniques would not be needed. If in any case we want to apply the technique, we can blur it once and apply histogram, which would finally cancel out the effects and give a positive result.

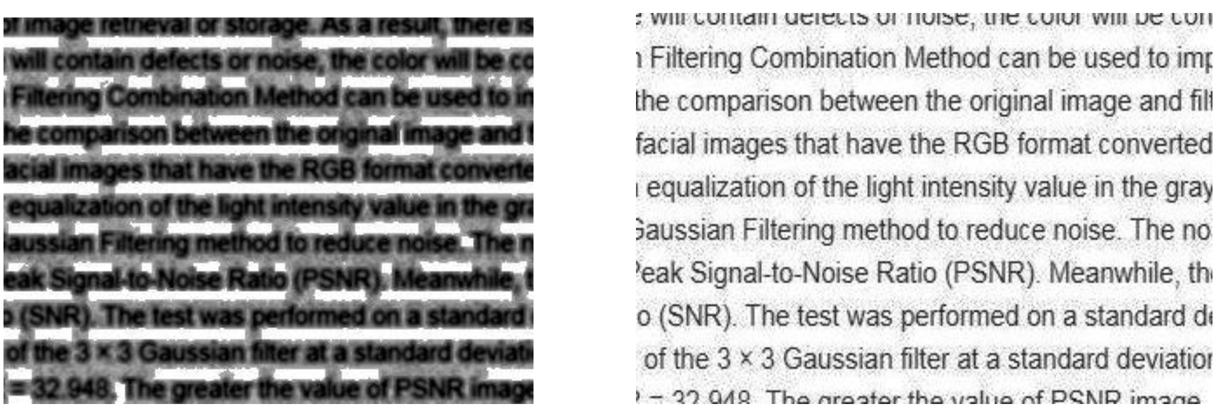



*Fig 22 : The 2 final steps.*

The image on the right is the final output after applying the adaptive PSNR method. Compare it with the original image.

The same rule can be applied to coloured images. They first need to be converted to HSV palette and the rule is applied only to the Hue or H channel. For the value or V channel, Gamma intensity correction must be applied. This would then be stacked and reverted back to BGR/RGB ( depending upon the used framework) palette.

# References

1. About Our Company - Silver Touch Technologies. (n.d.). Retrieved from https://www.silvertouch.com/about-us/
2. Alexander, Rosebrock, A., Peng, Sam, Abraham, N., Wilf, . . . Seed, L. (2020, April 18). Keras: Multiple outputs and multiple losses. Retrieved June 11, 2020, from https://www.pyimagesearch.com/2018/06/04/keras-multiple-outputs-and-multiple-losses/
3. Bansari, S. (2019, April 30). Introduction to how CNNs Work. Retrieved June 10, 2020, from https://medium.com/datadriveninvestor/introduction-to-how-cnns-work-77e0e4cde99b
4. Chollet, F. (n.d.). The Keras Blog. Retrieved June 11, 2020, from https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html
5. Dandynaufaldi. (2019, February 13). Dandynaufaldi/Agendernet. Retrieved June 23, 2020, from https://github.com/dandynaufaldi/Agendernet
6. Dwivedi, D. (2019, March 27). Face Detection For Beginners. Retrieved June 23, 2020, from https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9
7. Meade, R. (2020, January 14). Bias in Machine Learning: How Facial Recognition Models Show Signs of Racism, Sexism and Ageism. Retrieved June 23, 2020, from https://towardsdatascience.com/bias-in-machine-learning-how-facial-recognition-models-show-signs-of-racism-sexism-and-ageism-32549e2c972d
8. Papers with Code - Age Estimation. (2019). Retrieved June 23, 2020, from https://paperswithcode.com/task/age-estimation
9. Papers with Code - Age Estimation. (n.d.). Retrieved June 23, 2020, from https://paperswithcode.com/task/age-estimation
10. Parkhi, O. M., Vidaldi, A., & Zisserman, A. (n.d.). *Deep Face Recognition* (Deep Face Recognition).
11. Rcmalli. (2020, March 19). Rcmalli/keras-vggface. Retrieved June 11, 2020, from https://github.com/rcmalli/keras-vggface
12. Rcmalli. (2020, March 19). Rcmalli/keras-vggface. Retrieved June 23, 2020, from https://github.com/rcmalli/keras-vggface

13. Seif, G. (2019, May 04). Deep Learning vs Classical Machine Learning. Retrieved June 11, 2020, from https://towardsdatascience.com/deep-learning-vs-classical-machine-learning-9a42c6d48aa
14. Serengil, S., Says:, H., Says:, S., Says:, G., Says:, A., Says:, R., . . . Says:, A. (2020, June 18). Deep Face Recognition with VGG-Face in Keras. Retrieved June 23, 2020, from https://sefiks.com/2018/08/06/deep-face-recognition-with-keras/
15. Tf.keras.layers.SeparableConv2D : TensorFlow Core v2.2.0. (n.d.). Retrieved June 23, 2020, from https://www.tensorflow.org/api_docs/python/tf/keras/layers/SeparableConv2D
16. Uddin, M. S. (n.d.). Human Age Prediction from Facial Image Using Transfer Learning in Deep Convolutional Neural Networks. In J. C. Bansal (Ed.), *Proceedings of International Joint Conference on Computational Intelligence* (IJCCI 2019 ed., pp. 225-226).
17. VGG Face Descriptor. (n.d.). Retrieved June 23, 2020, from http://www.robots.ox.ac.uk/~vgg/software/vgg_face/
18. VGGFace2: A dataset for recognising faces across pose and age Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi and Andrew Zisserman Visual Geometry Group, Department of Engineering Science, University of Oxford {qiong,lishen,weidi,omkar,az}@robots.ox.ac.uk