Gender Detection using VGG Net

CSE 691: Machine Intelligence with Deep Learning

By

Narendra Katamaneni (SUID: 459715027)

Mihir Joshi (SUID: 299556995)

Saakshi (SUID: 328449779)

Akshitha Duddala (SUID: 571933326)

Under the Guidance

Of

Professor

Qinru Qiu

Department of Electrical Engineering and Computer Science

Syracuse University

Table of Contents

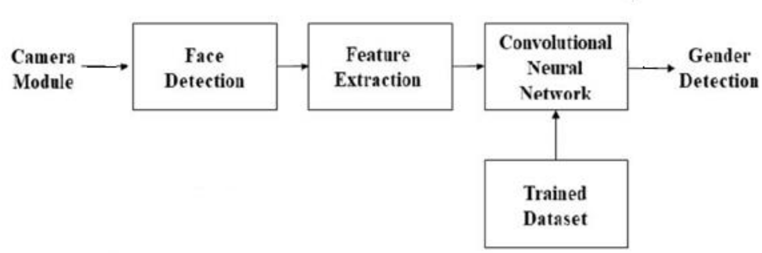
1)Introduction

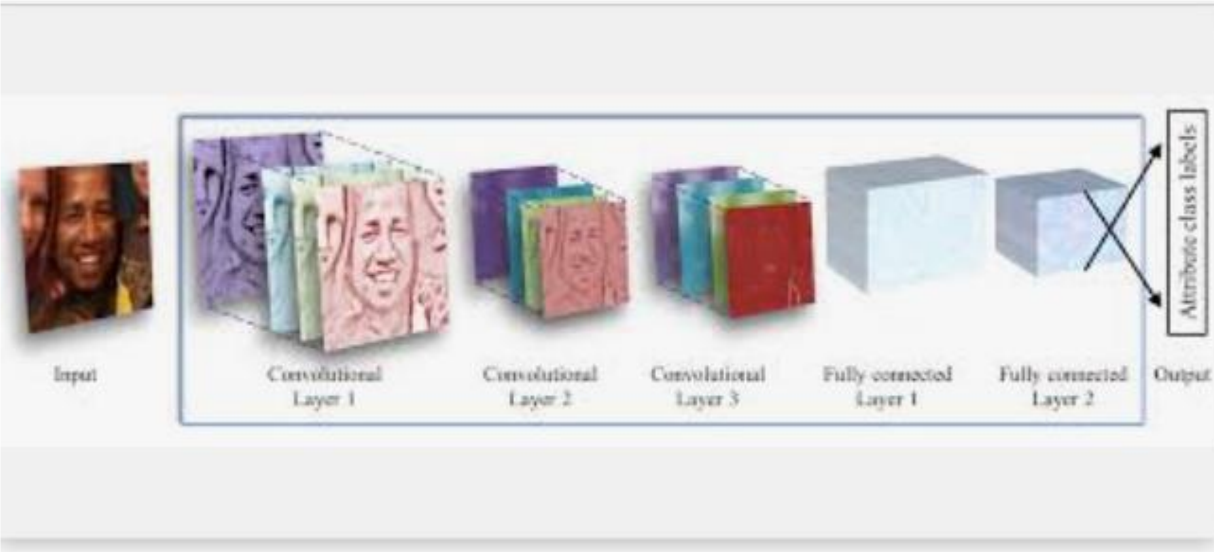
Automatic gender recognition has now pertinent to an extension of its usage in various software and hardware, particularly because of the growth of online social networking websites and social media. However, the performance of already exist system with the physical world face pictures, images are somewhat not excellent, particularly in comparison with the result of task related to face recognition.

An approach using a convolutional neural network (CNN) is proposed for real-time gender classification based on facial images. Gender classification was first perceived as an issue in psychophysical studies; it focuses on the efforts of understanding human visual processing and identifying key features used to categorize between male and female individuals. This became the basis for our classification. The CNN architecture that we use in our project exhibits a much-reduced design complexity when compared with other CNN solutions applied in pattern recognition. The number of processing layers in the CNN is reduced to only four by fusing the convolutional and subsampling layers. Unlike in conventional CNNs, we replace the convolution operation with cross-correlation, hence reducing the computational load. The network is trained using a second-order backpropagation learning algorithm with annealed global learning rates. We achieved an accuracy of 94% and the processing time on google cloud was approximately 20min. These results correspond to a superior classification performance, verifying that the proposed CNN is an effective real-time solution for gender recognition.

We have explored that by doing learn and classification method and with the utilization of Deep Convolutional Neural Networks (D-CNN) technique, a satisfied growth in performance can be achieved on such gender classification tasks that is a reason why we decided to propose an efficient convolutional network VGGnet architecture which can be used in extreme case when the amount of training data used to learn D-CNN based on VGGNet architecture is limited.

Block diagram of CNN is given below





The above figure shows the typical architecture of a convolutional neural network.

2)Application

Gender classification has become relevant to an increasing amount of applications, particularly since the rise of social platforms and social media. It is essential and critical for many applications in the commercial domains such as applications of human-computer interaction and computer-aided physiological or psychological analysis, since it contains a wide range of information regarding the characteristics difference between male and female. Gender is one of the key facial attributes, plays a very foundational role in social interactions, making gender estimation from a single face image an important task in intelligent applications, such as access control, human-computer interaction, law enforcement, marketing intelligence and visual surveillance, etc.

The development and progress in gender recognition technology has led to many potential uses in a large application scope, because the gender classification techniques can significantly improve the computer's perceptional and interactional capabilities. For example, gender classification can improve the intelligence of a surveillance system, analyze the customers' demands for store management, and allow the robots to perceive gender, etc. To be concrete, applications of automatic gender classification can be categorized in the following fields.

Human-Computer Interaction

Robots or computers need to identify and verify human gender to improve the customer experience based on personalized information. By successfully determining gender, the system can provide appropriate and customized services for users by adapting to them according to their gender.

Surveillance Systems

Classifying gender in surveillance systems for public places (e.g., bank, school) can assist intelligent security and surveillance systems to track moving objects, detect abnormal behaviors, and facilitate the security investigation of criminals who intentionally try to hide their identity information. In addition, gender-focused surveillance can help evaluate the threat level for a specific gender if the gender information can be automatically obtained in advance.

Commercial Development

Gender classification is useful for guiding effective marketing and establishing a smart shopping environment, in which production can be directed to specific users through websites, electronic marketing, and advertising, etc. For instance, in a supermarket or department store, knowing the number of male and female customers helps the store managers to make effective sales and managing decisions.

Demographic Research

The application of a gender classification system helps demographic research in efficiently collecting demographic information. Automatic identification of human gender enhances demographic statistics (e.g. gender, disability status, race definition) and population prediction. The ability to automatically detect gender information acts as a supplementary method to demographic research conducted in public places.

3)Architecture

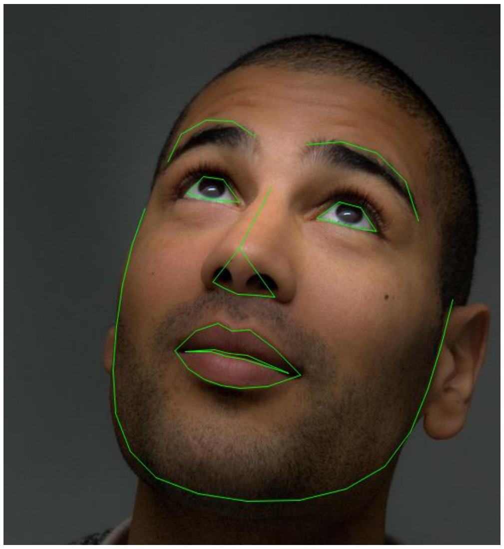
Architecture of our Convolutional Neural Network can be divided into two models.

a) Face Detection
b) Gender Detection

a) Architecture for Face detection

Facial Landmarks:

Face landmark detection is the process of finding points of interest in an image of a human face. It has recently seen rapid growth in the computer vision community because it has many compelling applications



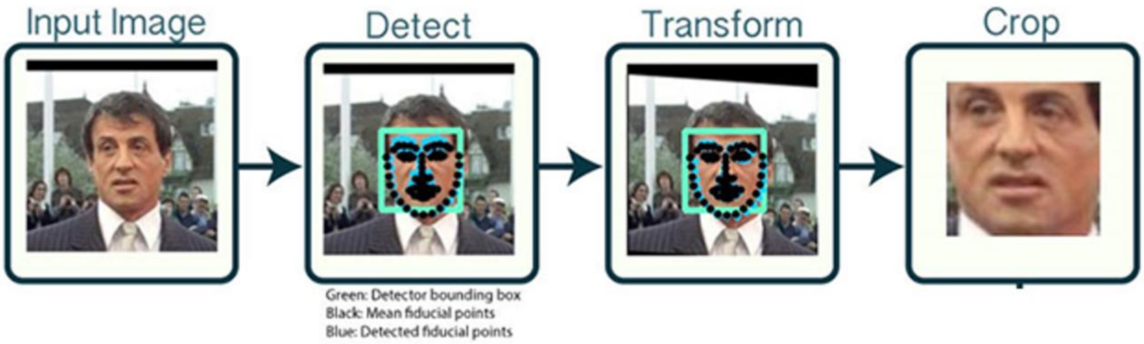
Detecting facial landmarks is a two-step process:

Step #1: Localize the face in the image.
Step #2: Detect the key facial structures on the face ROI

As once see in the image above, Key facial structures:

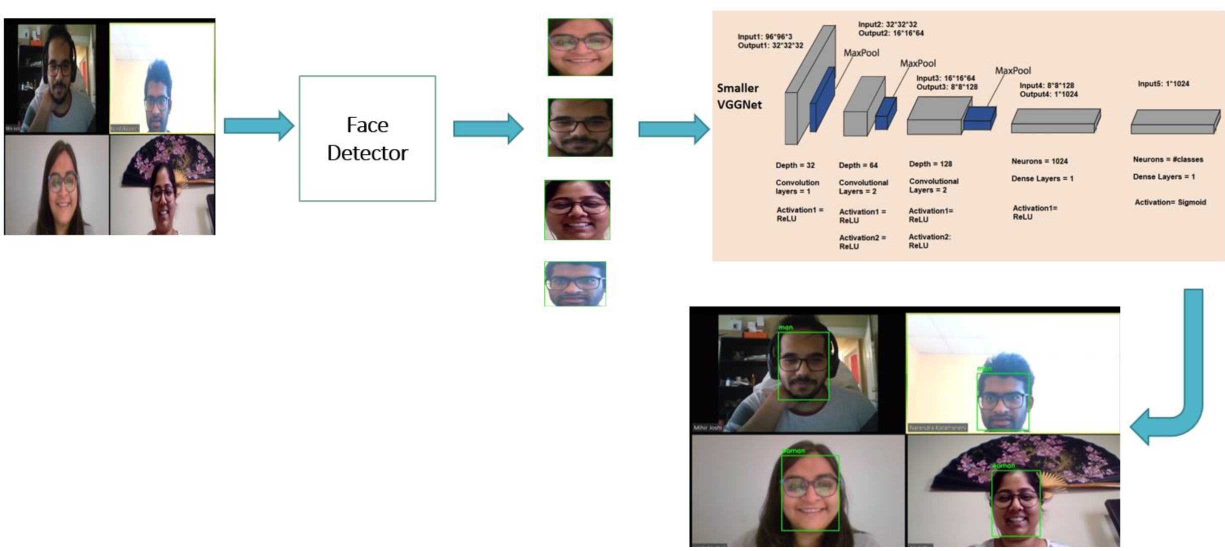
- Mouth
- Right eyebrow
- Left eyebrow
- Right eye
- Left eye
- Nose
- Jaw

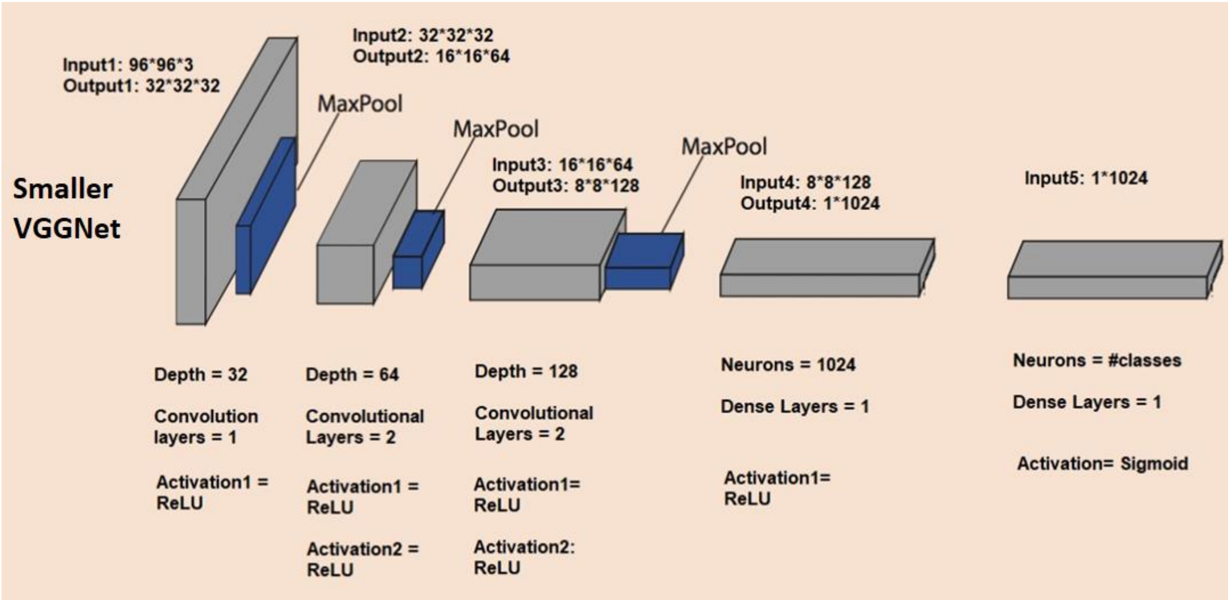
Efficient detection of facial landmarks is a vital step in the face detection system. As one can see in the below image which show the Architecture of Face detection system, given an input image, face detection algorithm tries to detect facial landmarks. Then the image is transformed and finally cropped.



b) Architecture for Gender Detection

Below Image shows various input and outputs for the entire gender detection neural network. Given an input image which consists of persons to a face detection system, it splits out the cropped images of all the human faces. Then, these cropped image/s are fed to the gender detection neural network as an input. Then, the convolutional neural network predicts the gender of all the human faces that are given as input to it.





Above figure shows the architecture of our smaller (Neural Network consists of fewer layers compared to a typical VGG16 network) VGGnet. Smaller VGGNet consists of 10 layers, were group of 2/3 layers are formed as a batch. Every batch consists of the batch normalization, max polling and drop out. ReLU activation function is used at almost every layer except for the final fully connected layer where sigmoid activation function is used.

4) Code implementation

Below is the list of packages that were used to implement the Convolutional neural network,

- Numpy
- opencv
- tensorflow
- keras
- requests
- progressbar
- cvlib
- scikit-learn
- matplotlib

Below code snippet shows the python implementation of the smaller VGGNet that was used in our project.

```
    model.add(Conv2D(32, (3,3), padding="same", input_shape=inputShape))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(MaxPooling2D(pool_size=(3,3)))
    model.add(Dropout(0.25))

    model.add(Conv2D(64, (3,3), padding="same"))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(Conv2D(64, (3,3), padding="same"))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.25))

    model.add(Conv2D(128, (3,3), padding="same"))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(Conv2D(128, (3,3), padding="same"))
    model.add(Activation("relu"))
    model.add(BatchNormalization(axis=chanDim))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.25))

    model.add(Flatten())
    model.add(Dense(1024))
    model.add(Activation("relu"))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))

    model.add(Dense(classes))
    model.add(Activation("sigmoid"))

# create groud-truth label from the image path
for img in image_files:

    image = cv2.imread(img)

    image = cv2.resize(image, (img_dims[0],img_dims[1]))
    image = img_to_array(image)
    data.append(image)

    label = img.split(os.path.sep)[-2]
    if label == "woman":
        label = 1
    else:
        label = 0

    labels.append([label])
```

Above code snippet shows the process that was used to apply labels to the training dataset. For simplicity all the woman images in the training dataset are labelled with ‘1’ and all the man images are labelled with ‘0’.

```
for idx, f in enumerate(face):
     # get corner points of face rectangle
    (startX, startY) = f[0], f[1]
    (endX, endY) = f[2], f[3]

    # draw rectangle over face
    cv2.rectangle(image, (startX,startY), (endX,endY), (0,255,0), 2)

    # crop the detected face region
    face_crop = np.copy(image[startY:endY,startX:endX])

    #cv2.imwrite("image{}.jpg".format(i),face_crop)
    i=i+1
    #preprocessing for gender detection model
    face_crop = cv2.resize(face_crop, (96,96))
    face_crop = face_crop.astype("float") / 255.0
    face_crop = img_to_array(face_crop)
    face_crop = np.expand_dims(face_crop, axis=0)

    # apply gender detection on face
    conf = model.predict(face_crop)[0]
    print(conf)
    print(classes)

    # get label with max accuracy
    idx = np.argmax(conf)
    label = classes[idx]

    label = "{}".format(label)
    Y = startY - 10 if startY - 10 > 10 else startY + 10
    # write label and confidence above face rectangle
    cv2.putText(image, label, (startX, Y),  cv2.FONT_HERSHEY_SIMPLEX,
                0.7, (0, 255, 0), 2)
# display output
cv2.imshow("gender detection", image)
```

Above code snippet shows the process that gender detection CNN applies for every cropped image that it gets as input from the face detection module. The final output is stored as “gender detection.png” image and is saved on the disk.

5)Learning and Testing

Dataset

We created our own Smaller VGGNet model from scratch by downloading ~4200 images from google; that is more than 2000 for each gender.

Below two images shows the nature of our data set for model to classify a human face as man

Below two images shows the nature of our data set for model to classify human face as woman

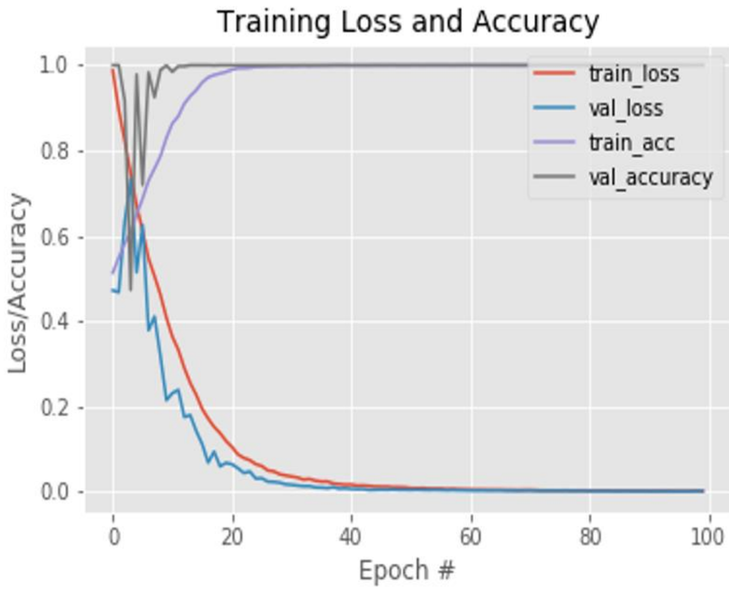
Our dataset is divided into Training and Validation datasets. 80% consists of Training Data and 20% Validation Data. We have taken random images from google to test out convolutional neural network.

Problem of Overfitting:

This is one of the common problems that every convolutional neural network face. This is the situation where; CNN makes accurate predictions for the validation data during the training but whenever a new unseen image is given to the model it fails to make accurate predictions.

we encountered the problem of overfitting on adding two extra layers with 256 kernels/filters each before the last two layers of the fully connected layers.

Below graph shows the loss/accuracy of the CNN model at an Epoch.

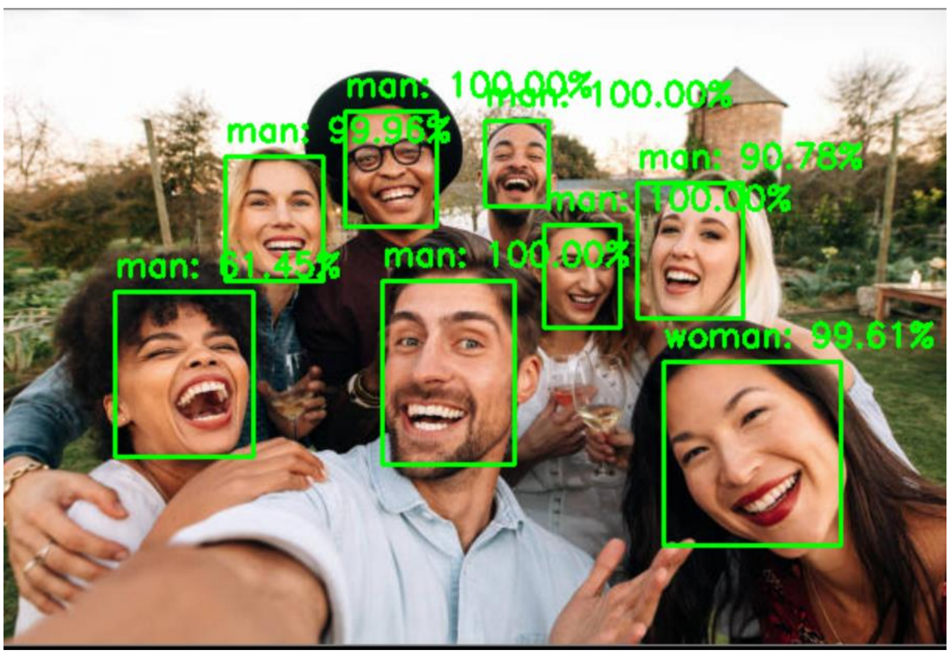


Overfitting

Added 2 convolutional layers of 256 depth

Learning rate : 1e-4
Epochs : 100

Train accuracy : 100%
Validation accuracy : ~98%



As one can see in the above image, even though the validation accuracy of our architecture is almost 100% during the training process, It was unable to predict the gender classification in the image, as it has predicted the woman as man for two faces on the left side of the image.

However, we can overcome the problem of overfitting by doing any/all the following steps

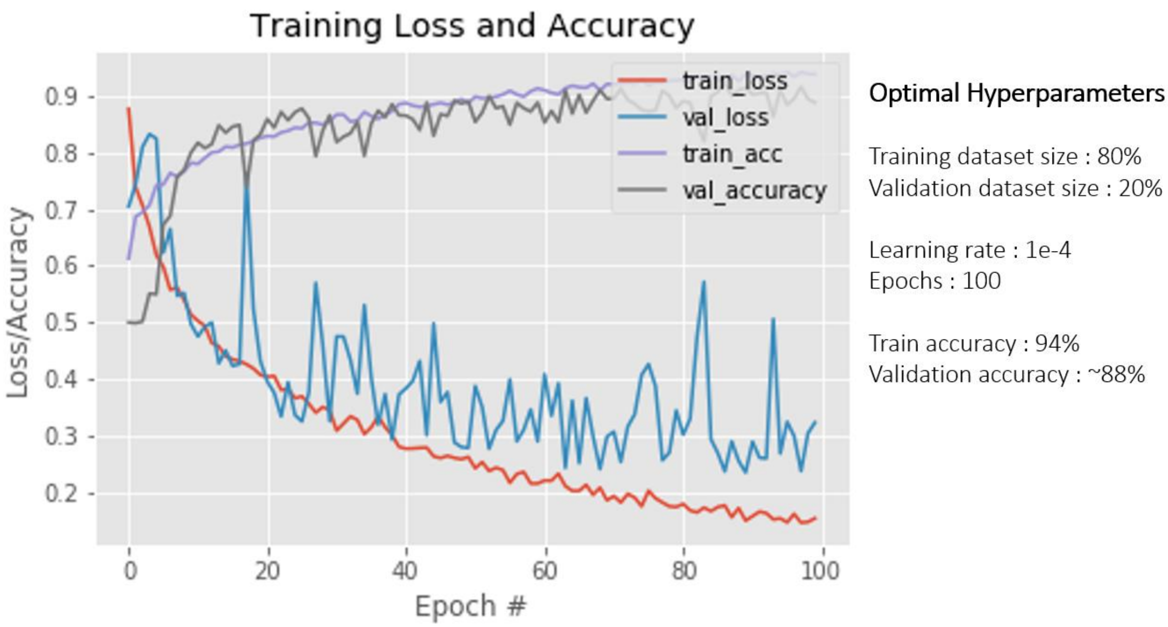
1.Increased dataset: Increasing the size of the dataset makes the CNN to update the weights more accurately and will be able to predict correctly when given a similar unseen image.

2.Data Augmentation: Data Augmentation itself can be considered as the process the increase the size of the dataset by performing various transformation on the images.

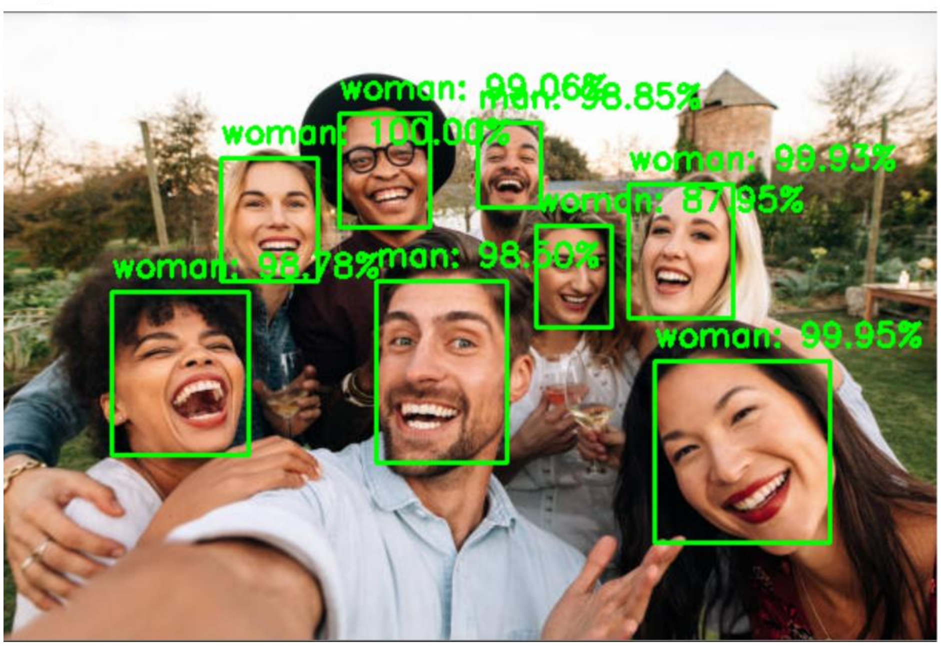
3.Depth of the network: Reducing the depth of model overcomes overfitting problem

4.Dropout: Dropout allows the network to train every neuron in the image more accurately thus increasing the chances of model predicting the data correctly.

In our model, we have followed almost all the above steps to avoid overfitting. Below graph shows the loss/accuracy for an epoch of our optimal neural network.



We can see from the above graph the optimal parameters which give the accurate results for test data. We can also see that the training accuracy and validation accuracy is increasing, and training accuracy is around 94%.



The above show the prediction of our model after overcoming overfitting problem. It is almost able to predict every human face in the image accurately.

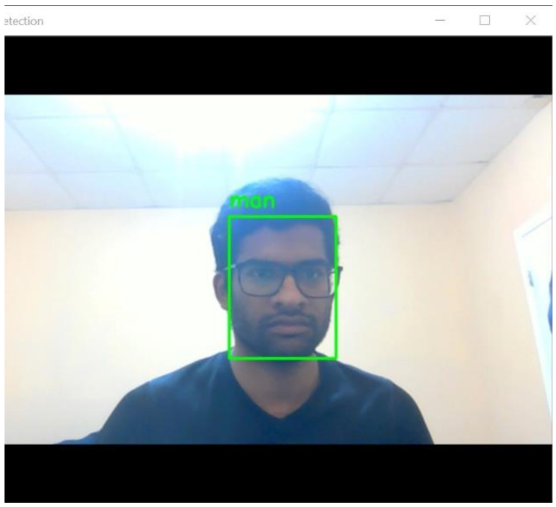
6) Integration with webcam

Integrated the gender detection model with the webcam to enable real time gender detection facility. While the webcam is turned on, every single frame is read from the webcam and is passed to the face detection system and the results are further passed onto the gender detection system to predict the face in the webcam.

Below code snippet shows the integration of CNN with webcam using python program

```python
# loop through frames
while webcam.isOpened():
    # read frame from webcam
    status, frame = webcam.read()
    if not status:
        print("Could not read frame")
        exit()
    face, confidence = fd.detect_face(frame)
    # loop through detected faces
    for idx, f in enumerate(face):
        # get corner points of face rectangle
        (startX, startY) = f[0], f[1]
        (endX, endY) = f[2], f[3]
        cv2.rectangle(frame, (startX,startY), (endX,endY), (0,255,0), 2)
        face_crop = np.copy(frame[startY:endY,startX:endX])
        if (face_crop.shape[0]) < 10 or (face_crop.shape[1]) < 10:
            continue
        # preprocessing for gender detection model
        face_crop = cv2.resize(face_crop, (96,96))
        face_crop = face_crop.astype("float") / 255.0
        face_crop = img_to_array(face_crop)
        face_crop = np.expand_dims(face_crop, axis=0)
        conf = model.predict(face_crop)[0]
        # get label with max accuracy
        idx = np.argmax(conf)
        label = classes[idx]
        label = "{}".format(label)
        Y = startY - 10 if startY - 10 > 10 else startY + 10
        cv2.putText(frame, label, (startX, Y),  cv2.FONT_HERSHEY_SIMPLEX,0.7, (0, 255, 0), 2)
    cv2.imshow("gender detection", frame)
    # press "Q" to stop
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
# release resources
webcam.release()
cv2.destroyAllWindows()
```

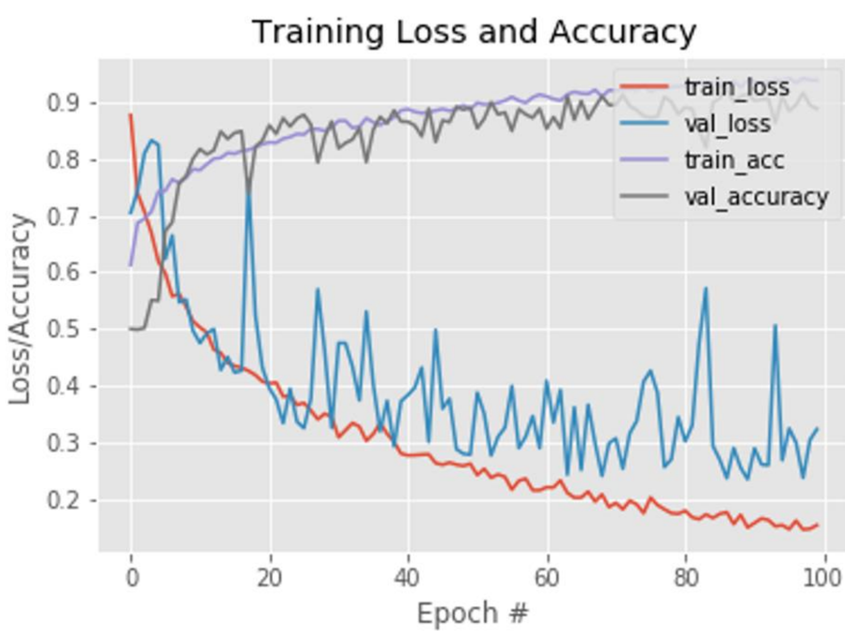
The sample output of a single frame from the webcam when passed through CNN is given below.



7)Analysis

The below graph shows the behavior of the model in terms of loss and accuracy at various epochs. The learning rate that we have used is 1e-4. As we can see from the graph, as the model kept training on the greater number of images (with every epoch), the model is able to validate the images more effectively thus reducing the validation and training loss by the end of training.

As the loss of the model kept decreasing for every epoch, both the training and validation accuracy of the model kept increasing gradually. We were able to get an accuracy of ~ 88% at the end of our training.



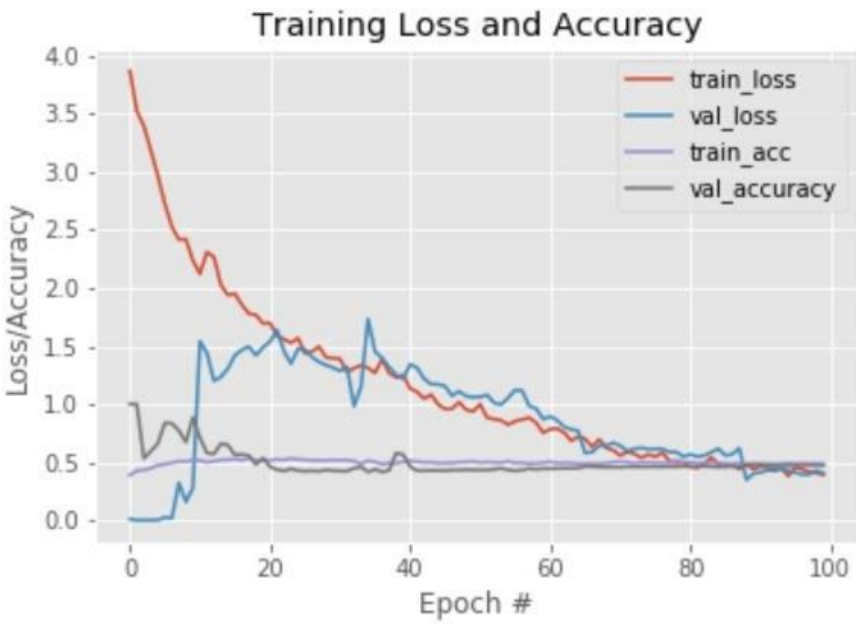
Optimal Hyperparameters

Training dataset size : 80%
Validation dataset size : 20%

Learning rate : 1e-4
Epochs : 100

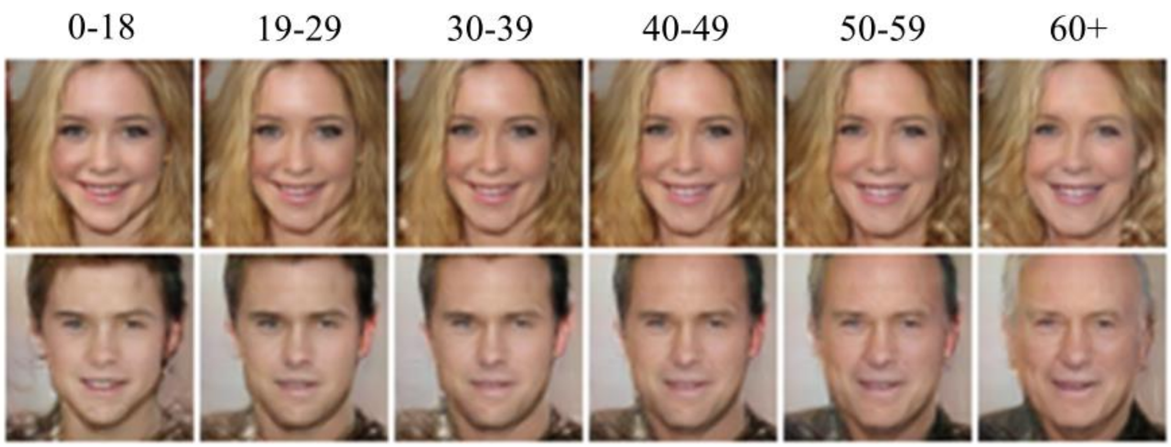
Train accuracy : 94%
Validation accuracy : ~88%

We have done various experiments by changing activation of the last output layer. Below graph shows the behavior of the model for the Tanh activation function in the last fully connected layer. Both the training loss and validation loss kept reducing till the end of the model. However, the validation accuracy and the training accuracy is not increasing as expected and shows almost a flat curve throughout the entire period of our training. And the accuracy that we are able to achieve is nearly 50%.



8)Extend and Improve

Although we have successfully performed gender detection using VGG Net and implemented it through a webcam, we believe that there is much scope for improvisation. For starters, we can expand our dataset to include more diverse images of faces for both, men, and women. Including more images of men and women of different age groups, having different facial structures will improve the efficiency of our model. There is evidently a lot of difference between the face of an adult as compared to that of an infant or a baby. A child's face lacks a defined jawline and a pointed chin. Another difference is that the spacing between the two eyes is less for children as compared to adults. These features of the children make it difficult to differentiate between a baby girl and a baby boy. Coming to older adults, their features seem to be hidden due to saggy skin and wrinkles. There is a loss of muscle tone and thinning in skin as we age, and this again makes it difficult to detect their gender. Increasing the dataset to include more images of people across varying age groups and varying facial structures can possibly improve our prediction using the increasingly trained model.



An area we wanted to expand our project into was diagnosis of diseases whose symptoms can be seen on the face. With advancement of technology every individual has access to resources at their home. One such facility is health care. There have been apps that help you diagnose a health issue when you enter the symptoms. Now especially, due to the COVID-19 pandemic, doctors have shifted to consultation over the phone or video call to treat minor health issues. Our project can be extended to help in such a situation. Changing the learning tags from man or woman to healthy skin and diseased skin we can detect whether a skin is infected or healthy. Secondly by changing the activation layer of the final stage from sigmoid to softmax can facilitate efficient learning with introducing various tags such as chicken-pox, eczema, measles etc. so as to not just detect diseased or healthy skin but also to differentiate the disease of the afflicted patient.

9)Summary

Despite numerous past strategies, we must manage the issue of gender recognition through face image, in this project we have set some standards which rely on the very well-known VGGNet network architectures. We tried to implement gender recognition through face image as well as through webcam by tweaking the original VGGNet architecture into a smaller version of it to avoid overfitting. We have majorly used OpenCV, keras, tensorflow python library to implement the code on a Google Cloud GPU giving us a great result on a large dataset. The most troubling part of this project was to find the optimum number of layers to avoid overfitting which was occurring in the original VGGnet.

10)References

1. Yujie Dong and Damon L Woodard. Eyebrow shape-based features for biometric recognition and gender classification: A feasibility study. In Biometrics (IJCB), 2011 International Joint Conference on, pages 1–8. IEEE, 2011.
2. Abdenour Hadid and Matti Pietikäinen. Combining appearance and motion for face and gender recognition from videos. Pattern Recognition, 42(11):2818–2827, 2009.
3. Feng Lin, Yingxiao Wu, Yan Zhuang, Xi Long and Wenyao Xu. Human Gender Classification: A Review, 2012.
4. Tivive FHC, A. Bouzerdoum (Sep 2006) "A gender recognition system using shunting inhibitory convolutional neural networks" In: International Joint Conference on Neural Networks; Vancouver, Canada. New York, NY, USA: IEEE. pp.5336-5341.
5. Eidinger R. Enbar, T. Hassner, "Age and gender estimation of unfiltered faces" IEEE Transactions on Information Forensics and Security, 9 (12) (2014), p. 21702179.
6. Dong Chen, Xudong Cao, Fang Wen, Jian Sun (2013)," Blessing of Dimensionality: High-dimensional Feature and Its E cient Compression for Face Verification", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3025-3032.
7. Y. H. Kwon and N. da Vitoria Lobo (Jun 1994) "Age classification from facial images "In Computer Vision and Pattern Recognition, 1994. Proceedings CVPR., 1994 IEEE Computer Society Conference on, pages 762767
8. Gkhan zbulak, Yusuf Aytar, Hazm Kemal Ekenel (September, 2016.) "How Transferable are CNN-based Fea-tures for Age and Gender Classification", arXiv:1610.00134v1 [cs.CV].