Week:6 ADML Assignment Augmentation & Transfer Learning

Initial data visualization and spotting data problems

For this assignment we were working with an image set of sign language for few alphabets, initially I plotted some of these images into a 2 by 5 layer as seen in fig. 1, on initial glance we can spot a lot of issues with these images, they are illumination, rotation, position of hand, zoom, scale, etc.

Fig.1 Hand Signs for G

Archetype of MobileNet V2:

MobileNetV2 is a convolutional neural network architecture that is designed for mobile and embedded vision applications. It is based on an inverted residual structure where the input and output of the residual block are thin bottleneck layers. This can be seen in the Fig.2 where the general architecture of MobileNetV2 has been described. This contrasts with traditional residual models which use expanded representations in the input. MobileNetV2 uses lightweight depth wise convolutions to filter features in the intermediate expansion layer. Additionally, non-linearities in the narrow layers were removed to maintain representational power.

The MobileNetV2 model has been pre-trained on the ImageNet dataset. ImageNet is a large dataset of images, which is commonly used for training deep learning models. The ImageNet dataset contains over 14 million images, and each image is labelled with one of 1000 different categories.

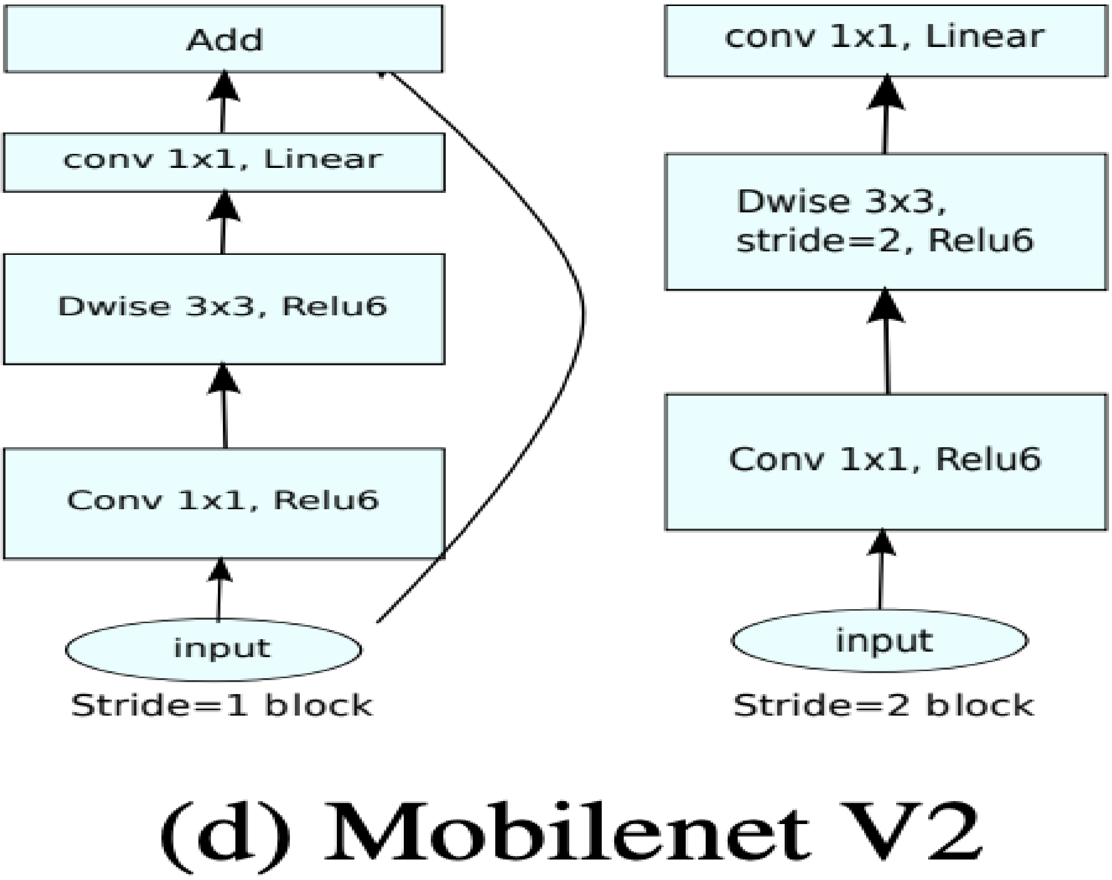

Fig.2 Mobilenet V2 Architecture

Data Augmentation:

Data augmentation is a strategy that can significantly improve the performance of machine learning models, especially when the amount of available training data is limited. It involves creating new training samples by applying transformations to the existing data. These transformations can include rotations, translations, scaling, flipping, cropping, and more. The goal of data augmentation is to create a more diverse training set that can help the model generalize better to unseen data.

```python
train_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')
```

One of the main issues that data augmentation can help address is overfitting. Overfitting occurs when a model learns the training data too well, to the point where it performs poorly on unseen data. This is often because the model has learned to fit the noise in the training data, rather than the underlying pattern. By creating a more diverse training set with data augmentation, we can help the model generalize better and reduce overfitting.

Another issue that data augmentation can help address is the curse of dimensionality. As the dimensionality of the data increases, the volume of the data increases exponentially, making it more difficult to sample from the data space. Data augmentation can help mitigate this issue by creating more training samples with the same amount of data.

In our case we are using various parameters for Image Augmentation as seen from the above code snippet, width shift, shear range, zoom, horizontal flip, rotation being few of them.

Importance Of Augmentation:

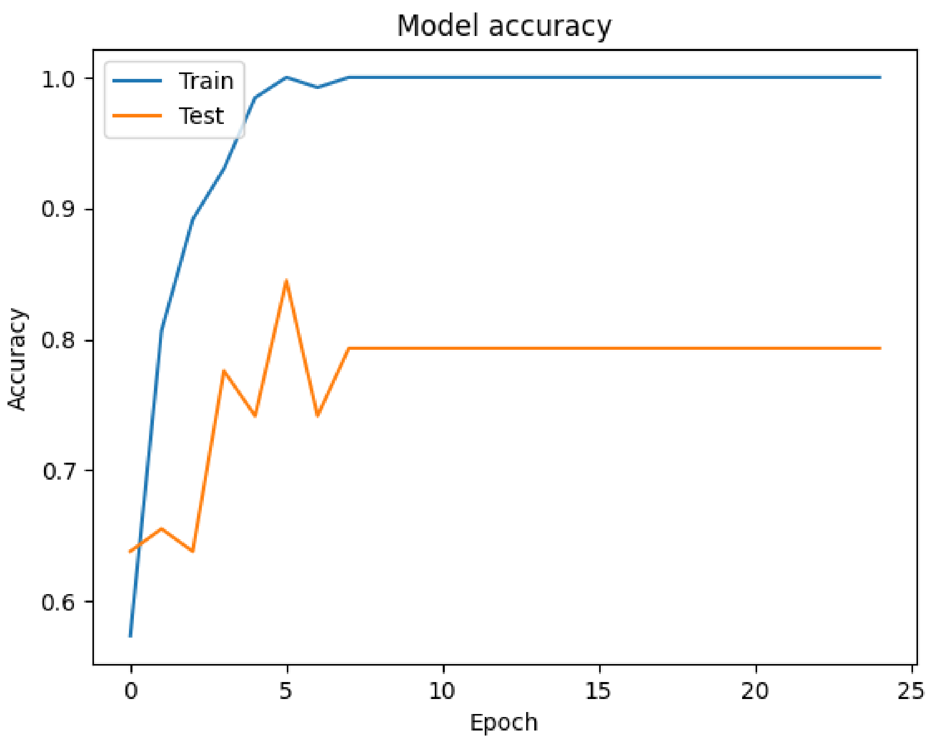

Fig.3 Model Accuracy for Non Augmented Images.

As discussed before, augmentation solves the issue of overfitting, among others. If we look at Fig. 3, we can see that the model performs extremely well for training images, with almost 100% accuracy. However, when it comes to the test dataset, the accuracy drops to 80% and flattens, illustrating a classic example of overfitting.

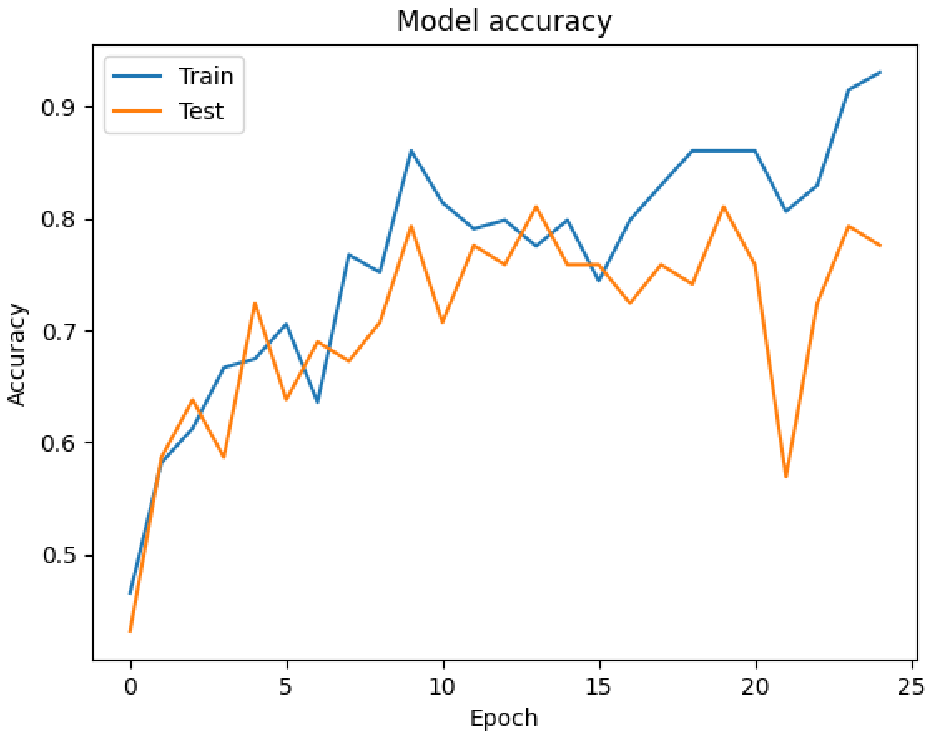

Fig.4 Model Accuracy for Augmented Images.

Checking Fig. 4, we observe that the model's accuracy is ever-fluctuating. Even at 25 epochs, although the model performs well on the training set compared to the testing set, it is still trying to compensate.

Model Performance on My Images:

I used this model to predict some of the images that I employed, and the results were optimistic, to say the least, as seen in Fig. 5. It performed well in predicting 'M,' which consists of three fingers, correctly identifying 1 'H' and 1 'G.' However, for the rest, it struggled to predict accurately. This discrepancy could be attributed to the model being trained on images with similar hand sizes, shapes, and skin colours. For this task, these factors could play a significant role, as the training hand differed significantly from mine—being smaller and having a fairer complexion.

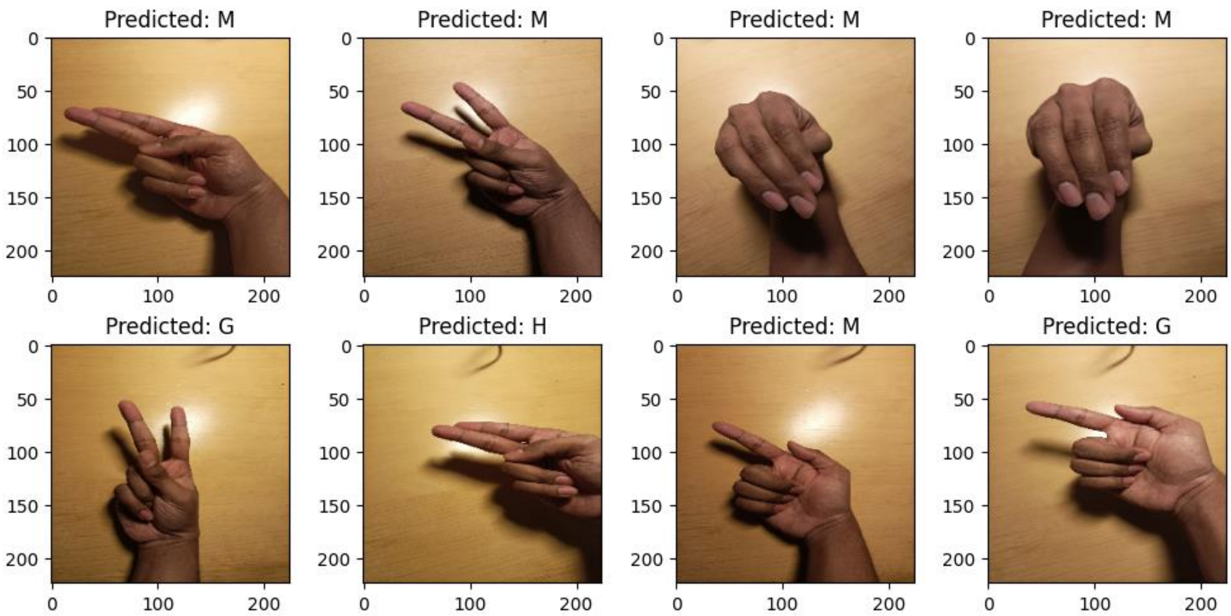

Fig.5 Personal Images Prediction