**Report on Road Classification**

**Objective:**

1. professor ask to build image classifier which will classify roads based on whether it is normal or damaged. Told me to collect around 100 images of normal roads and roads with potholes separately.

Then discuss about all the hyperparameters used in creating and training model. What are the different metrics used in model?

**Image Classification:**

Image classification pipeline can be formalized as follows:

* Input dataset named as Road\_images contains 200 images of normal roads and roads with potholes in separate folder named as normal\_new and potholes\_new.
* Then, I used that training set train the classifier to learn what every one of the classes looks like.
* At the end, I evaluated the quality of the classifier by applying prediction on the separate folder pre\_data and after further calculation applied confusion matrix over predicted data.
* Creation of model:
* I have created 6 Layers convolutional Neural Network for better accuracy.
* CNN Sequential model, I have used here 2D convolution layer, this layer created a convolution kernel that is convolved with the layer input to produce a tensor of outputs.

When using first layer in a model, provide the keyword argument input\_shape as tuple of integers, which does not include the batch axis. I have taken it as (125,125,3).

* Arguments used in the model:
* **Filters**: Means the total no. of filters in the convolution and its in the integer format, here in our case it is taken as 200
* **Kernel**\_**size**: it can be an integer or tuple/list of 2 integers, which tells the height and width of the 2D convolution window, basically standard value of kernel\_size is (3,3), so I have taken (3,3) in it.
* **Activation**: we need to apply activation function to the output of our CONV layer to make it linear. In our case it ‘relu’

So, the ReLU (Rectified Linear Unity) activation function simply changes all the negative value to 0, while leaving the positive values unchanged.

* **Pooling**: It’s the process where we reduce the size or dimensionality of the Feature Map.

Purpose of pooling to reduce the number of parameters to train the model. by retaining the most important features.

There are 3 types of pooling: MAX, SUM, Average

In our case we used MAX pooling and the value is (5,5) with stride 5, what it simply does it divide the feature map into 5 X 5 matrix and extract the maximum value from it.

Pooling reduce the dimensionality of the feature map by half and thus remove 75% of the activation seen in the previous layer.

All these 4 arguments make 1 layer of the CNN model and I have used here 6 layers of convolution layer.

* In last layer of CNN network, we use Activation Function or final activation function as SoftMax function. Which tell the output of the fully connected layer, FC layer output the class probability, where each class is assigned a probability and all probabilities must sum to 1. In our case there are 2 classes so the probability will be divided into 2.
* **Flatten:** it is used to convert the 2D into 1D form.
* **Dropout:** Prevent a model from overfitting

After creating the model I have tried it to save as image in the directory but it giving the error as ‘InputLayer’ object is not iterable.

* Compilation is the most important part for the model.
* So, during creation of model I Compiled the model using the arguments:
* **Optimizer**: Adam Optimizer

It is an algorithm for first order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments.

I have used it because it is straightforward to implement and is computationally efficient. It is useful for data full of noises. It requires little memory and invariant to diagonal rescaling. Have taken learning rate as 0.0001

* **Loss** **Function**: sparse\_categorical\_crossentropy

It basically used when we need to train the model containing dataset divided in different categories. So, for categories there are two loss functions as categorical cross entropy and sparse categorical cross entropy. Sparse categorical entropy uses a single integer for a class, rather than a whole vector which reflects the effect as it saves time in memory as well as computation.

This is the advantage of sparse categorical crossentropy over categorical entropy as in it uses the whole vector.

* **Metrics used**: Accuracy

Accuracy metric is used for evaluation of the model that tell the performance of the model, for classification model it is the most common type of metric, but it only gives satisfying result. As our model needs equal no of samples belonging to each class normal road and road with potholes. That’s the reason I have used it to compile the model.

Accuracy in this case is much better, just because I have the standard dataset which is created from the dataset available on Kaggle for pothole detection

**Accuracy value** **in this case**:

Accuracy on training set containing 200 images: 90.71%

Accuarcy on validation set which is the 30% of original dataset of 200 images: 90%

After that I have plotted the graph of model accuracy and model loss:

**Model Accuracy**: The graph showing Accuracy per epoch value:

A graph which shows the accuracy of training and validation set or testing set for each epoch value.

**Model Loss:** The graph showing loss per epoch value:

A graph which shows the loss of training and validation set for each epoch value.

**Prediction:**

Now it’s the time for evaluating the **Prediction** for certain dataset which is different from the training dataset. I have created the data set of 40 images for prediction which have same dimension of 125 X 125 as of training dataset.

Prediction is also the best way of evaluating the accuracy of the model and it also tells that well our model can predict other images for the certain classes.

Now, let’s move on the other evaluation of the performance of the model and the evaluation which play a very important role in classification problems i.e., **Confusion** **Matrix**.

**Confusion Matrix:**

Confusion matrix allows the visualization of the performance of an algorithm. It is also known as an error matrix. It allows easy identification of confusion between classes.

It’s the summary of prediction results on problem, no. of correct and in correct predictions are summarized with count values and broken down by each class.

Like in our model:

0 represents normal\_new class

1 represents Potholes\_ new class

00 represents predicted normal

11 represents predicted potholes

The Confusion matrix obtain by these classes are like:

A close up of a clock

Description automatically generated

So, According to Confusion matrix.

* True Positive (TP): 14 -(0,00): It means normal is predicted true for 14 times
* False Negative (FN): 6 – (0,11): It means normal is predicted potholes for 6 times
* True Negative (TN): 19 – (1,11) It means potholes is predicted potholes for 19 times.
* False Negative (FP): 1 – (1,00): It means potholes is predicted normal for 1 times

Now, here I discussed about some results based on the confusion matrix are as:

**Classification Rate/Accuracy**:

Accuracy= (TP+TN)/ (TP+ TN+ FP+ FN)

Accuracy= (14+19)/(14+6+1+19) = 0.825

**Recall**:

Defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples.

Recall= TP/(TP + FN)

Recall= 14/(14+6)= 0.7

**Precision**:

High precision indicates an example labelled as positive is indeed positive.

Precision= TP/(TP+ FP)

Precision= 14/ (14+1)= 0.9333

**F-** **measure**:

F-measure: (2\* Recall\* Precision)/(Recall + Precision)

F-measure= (2\*0.7\*0.93)/(0.7+0.93)= 0.7987

1. After that since the image classifier is not giving correct prediction on certain images which are blurred and taken from certain other angles. So, professor asks me to create a new model which will take into account all these scenarios and give better results on those images. Develop an image augmentation pipeline using standard libraries and train a new model and perform inference.

Share the source code along with other details of how you decide on which augmentation techniques needs to be used.

According to the situation of problem and for the better accuracy of the model, we must apply image augmentation.

**Image Augmentation:** Image augmentation has been developed in order to generate training data from an existing dataset. It is the process of taking images that are already in a training dataset and manipulating them to create many altered versions of the same image this both provides more images to train on, but can also help expose our classifier to a wider variety of lighting and colouring situations so as to make our classifier more robust.

As per requirement of our model since model doesn’t gave the correct prediction result for blurred and images taken from different angles.

For different angle we will use augmentation method of rotation to different angles.

And for removing blurring I have very less knowledge about it and don’t know the exact method for it, but I have tried to implement the data augmentation method on images but unfortunately failed to implement.