**Group 55: Data Science Challenge**

**Group members:** Rahul Rao Boinapally, Rana Ali Azeem, Ehsanullah Tabassum.

**Introduction**

The galaxy picture we were provided as part of the data science challenge contains both fake and real examples of galaxy. The image is divided into training and validation set with 3000 images in each class of training set and 1000 image of each class in validation set.

**A collage of images of galaxies

Description automatically generated with low confidence** **A picture containing screenshot, astronomy

Description automatically generated**

Figure 1:sample of real images sample of fake images.

This is a binary classification problem where a neural network must decide whether the image of galaxy is fake or real. We use data augmentation for photos by randomly rotating to expand the amount of the dataset, and used median filter to smooth is image and improve the clarity of the image. We have proposed a CNN network with batch normalization and dropout layer. The design of the project is as follows.

A picture containing text, screenshot, font, electric blue

Description automatically generated

Figure 2: design of the project.

**Implementation**

For the development, I have used Python and its libraries like OpenCV, TensorFlow, Matplotlib, and NumPy. The median filter is used in OpenCV, and the neural network is developed using TensorFlow. For measuring the performance of the model, I have used accuracy and binary cross-entropy loss as metrics. The neural network that is designed is made up of nine layers: two convolutional layers, which help in finding the features of the image; two batch normalisation layers, which help in faster and more stable training; two average pool layers, which reduce the size of feature maps; one dropout layer, which helps in regularisation of the neural network; one flatten layer, which helps in providing proper input to the FNN layer; and lastly, two FNN for classification.

Looking at Figure 1, we have observed that the images that are provided are noisy and need to be smoothed; hence, a median filter is applied. When it comes to training neural networks, they always require more data to train, so I have used data augmentation methods to add samples of data for training. For the median filter, I have used a window size of 3 for applying the filter to the image, and for data augmentation, I have used a parameter of rotation of 45, which randomly rotates the image by 45 degrees.

A picture containing text, screenshot, astronomical object, astronomy

Description automatically generated

Figure 3:outpuut of image augmentation and median filter.

CNN network used for the project:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 4: proposed architecture of CNN.

The layers and its parameters are explained in the following table.

A picture containing text, screenshot, font, number

Description automatically generated

**Results.**

I have trained the model for 4 epochs, with batch size of 32, metrics as accuracy, optimizer as Adam and loss as binary cross entropy.

A picture containing text, screenshot, line, plot

Description automatically generated A picture containing text, screenshot, line, diagram

Description automatically generated

Figure 5: accuracy of the model while training. Loss of the model while training.

From the above graphs, we can see that the model was not overfitted and was very stable during the training. The model was able to achieve a score of 99.7% accuracy on the training set and 98.85% accuracy on the validation set at the end of training. The loss was almost zero at the end of training on both datasets. The performance of the model on the validation set was lower than the training set during the first two epochs but showed significant improvement during training. We have also observed that the model took only 4 epochs to get this score, which was very fast. The epochs also took only 12 seconds to train, and the total training time was around 48 seconds. This shows the effect of the batch normalisation layer on faster training. The model was also able to capture the features of the image at a faster rate as we applied a median filter to the image and used data augmentation, which gave us more examples to learn from.

**Reference:**

Thanh, D.N.H. and Engínoğlu, S., 2019. An iterative mean filter for image denoising. *IEEE Access*, *7*, pp.167847-167859.

Albawi, S., Mohammed, T.A. and Al-Zawi, S., 2017, August. Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET) (pp. 1-6). Ieee.

Developers, T., 2022. TensorFlow. Zenodo.