Identical Meme Classification: A Neural network approach

AI Image Recognition Challenge

Aditya Shetty 21200138 and Teja Bhat 21201067

The project report is submitted to University College Dublin
in part fulfilment of the requirements for the degree of
MSc Data and Computational Science

School of Mathematics and Statistics
University College Dublin

Supervisor: Dr. Sarp Akcay

July 31, 2022

1 Abstract

Automatic understanding and analyzing of identical images by the system is difficult as compared to human visions. Several research has been done to overcome problems in existing classification systems, but the output was narrowed only to low-level image primitives. However, those approaches lack accurate classification of images. With the rise and expansion of the internet, memes have led to an increase in the number of interesting images such as dogs or bagels, muffins, Chihuahua, and many more which are much alike. These kinds of memes led to challenges in terms of classification. To classify these challenging memes, a model is built using a convolution neural network (CNN), a deep learning algorithm. The model can identify whether the image has a Chihuahua or muffin, and a dog or bagel based on several features of the images. The algorithm is written in several steps such as image filtering, dimension reduction of features or pooling, padding, and flattening operations. A large amount of data is created using data augmentation to get better accuracy. About 90 percent of the accuracy is generated by the model for the classification of similar (bagel or dog, muffin, or Chihuahua) images. CNN model is helpful to detect the differentiate the alike images learning and using different features of images and works without any human supervision.

2 Introduction

In recent years, due to the explosive growth of digital content, the automatic classification of alike images has become one of the most critical challenges in visual information indexing and retrieval systems. Computer vision is an interdisciplinary subfield of artificial intelligence that aims to give the similar capability of humans to computers for understanding information from images. In the case of humans, understanding the alike images and classification is a very easy task, but in the case of computers, it is not like humans. There have been breakthroughs in image labelling, object detection, and image classification, reported by different researchers worldwide. This leads to making it possible to formulate approaches concerning object detection and classification problems. Since image data is very different from tabular data, a special kind of neural network is used to deal with its complexities and derive insights from it, known as Convolutional Neural Networks Artificial neural networks have shown a performance breakthrough around object detection and alike image classification, especially Convolutional Neural Networks (CNN), this network focuses on identifying the image which almost looks like another image. [13]

Feature extraction is a key step of such algorithms. Feature extraction from images involves extracting a minimal set of features containing a high amount of object or scene information from low-level image pixel values, capturing the difference among the object categories involved. CNN is used not only in image classification. But also, it has its huge contribution to the Healthcare system, Digitalization of Identity cards, Signature forgery, etc. And is another method that can be used to obtain a higher discrimination capability in various classification problems such as Support Vector Machines. The CNN outperforms the SVM classifier in terms of testing accuracy. In comparing the overall correctives of the CNN and SVM classifier, CNN is determined to have a static-significant advantage over SVM

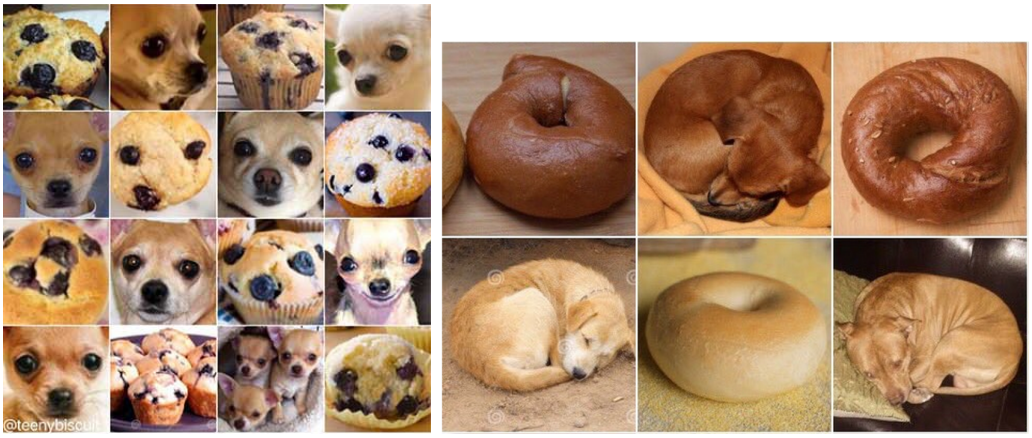

Figure 1: Goofy identical images

when the pixel-based reflectance samples are used, without the segmentation size. In this paper, we use a deep learning model for alike image classification, i.e., Convolution Neural Network.

3 Literature Review

Before the existence of smart phones and artificial intelligence concepts, people performed image classification using tools such as OpenCV. Since the emergence of artificial intelligence, deep learning, and the internet at large, researchers developed a wide variety of models for image processing. As the images got sophisticated, deep learning methodologies such as Convoluted Neural Networks showed impressive performance in the image classification problem. AlexNet based on deep learning model CNN in 2012, which won the championship in the ImageNet image classification of that year, and deep learning began to explode. In 2013, Lin et al. proposed the network in network (NIN) structure, which uses global average pooling to reduce the risk of overfitting. In 2014, GoogLeNet and VGGNet both improved the accuracy on the ImageNet dataset. GoogLeNet has further developed the v2, v3 and v4 versions to improve performance. Convoluted Neural Networks (CNN) were developed to evaluate visual information. The model tends to create similar feature values from local regions with similar patterns. In the process of realizing the image classification, CNN can use handcrafted characteristics of pictures, such as information on the edge and the distribution of the colours, and then separate the different kinds of objects. [8] [11]

4 Software tools and Libraries

Google colab Notebook

- Google colab is based on Jupyter open source allowing to create and share the file without installing anything.

- Google colab is allowing to write and execute the python code without having a local setup.

Keras Library: An open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. In addition to standard neural networks, Keras has support for convolution and recurrent neural networks. It supports other common utility layers like dropout, batch normalization, and pooling

5 Methodology

Convolution Neural Network is not very different from the original neural network. Like any other neural network, it is also made up of an input layer, one or more hidden layers, and an output layer.

5.1 Data Augmentation

Since CNN expects huge data to improve performance and data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images from the existing data. It can be done in several ways, such as:

- rotation_range: to randomly rotate images through any degree
- height_shift_range for a vertical and width_shift_range for a horizontal shift of the image.
- shear_range specifies the angle of the slant in degrees and zoom_range to randomly zoom the inside picture
- parameters horizontal_flip and vertical_flip for flipping along the vertical or the horizontal axis.

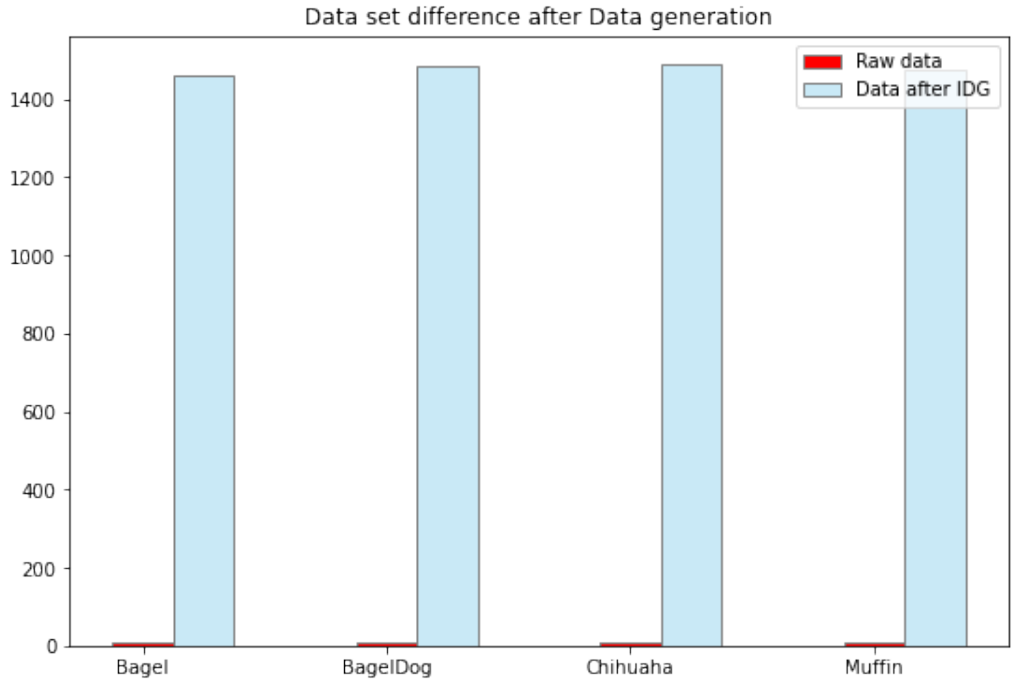

Figure 2: Four different categories of data before and after the data augmentation

Figure 2 explains the number of existing data and the data generated after the data augmentation. It increased from around 15 images to above 1500 images. Generated images are stored for visualization.

5.2 Splitting up the data

Data contain 4 categories with the images of Bagel, BagelDog, Muffin and Chihuahua. Each category is divided into training, validation and testing the model in a specified ratio. 60 percent of the data is to train the model, the remaining 40 percent is divided equally for the validation and testing. Figure 3 explains the data splitting. From the graph, most of the data can be seen in training set for all four different categories of data. Validation and Testing sets are having few data.

5.3 Model Build

- Convolution layer: The convolution layer is the first computing layer and core component of CNN used to extract the features from the input image. Images are often static, and non-linear in nature and consist of the smallest feature elements arranged in a two-dimensional grid and each element is presented by a pixel. It contains a set of learnable filters called convolution filters which slide over the input image. Each tiny section of the image passes through step by step to complete the feature map.[1].

- The number of filters is the number of neurons since each neuron performs a different convolution on the input to the layer. Initially, when the number of filters is 16, it

Figure 3: Four different categories of data distribution for training, validation and testing

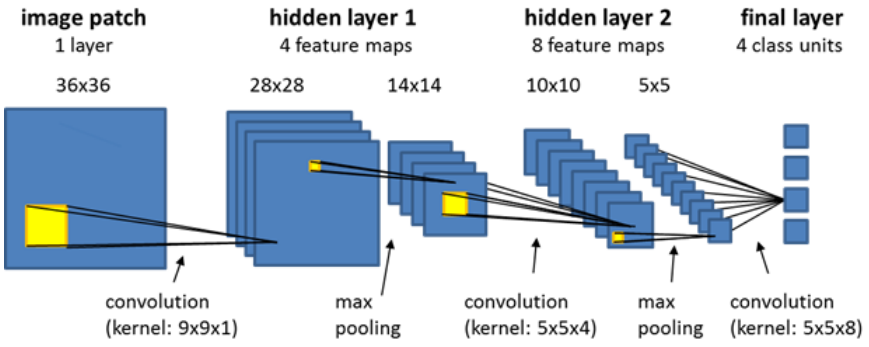

Figure 4: Architecture of Convolution Neural Network

considers the whole image where features are not specific. As the number of filters increases to 32, 64, and 128 more features can be extracted and a deeper, complicated network is built [1] [9]

$$W2 = \frac{W1 - F + 2 * P}{S} + 1 \tag{1}$$

$$H2 = \frac{H1 - F + 2 * P}{S} + 1 \tag{2}$$

$$D2 = K \tag{3}$$

Convolution layer accepting a volume of size [WI×HI×D1] where W1 is the width, H1 is the height and D1 the depth, the outputs of neurons in this type of layers are calculated by applying the product between their weights and a local region they are

connected to in the input volume. The obtained output volume [W2×H2×D2] called convolution maps. [7]

F:Spatial extend of the filter.

K:Number of filters.

P:Zero padding

S:Stride

- A Relu (Rectified Linear Units) activation function performs the function y = max(x,0), so the size of input and output in this layer are similar. It increases the non-linear features of the decision function and of every network without assuming receptive fields of the convolution layer.

$$y_i = \begin{cases} y_{a,i} & \text{for} \quad y_{a,i} \geq 0 \\ 0 & \text{for} \quad y_{a,i} < 0 \end{cases} , \tag{4}$$

Equation 4 explains the ReLu function with yi as the output from the activation function while ya,i is its input. ReLU uses an identity map in the positive part. This special design alleviates the vanishing gradient problem and accelerates the training process so that we can train very deep CNNs. [15].

- The SoftMax Function is a generalization of the Logistic Function, and it makes sure that our prediction adds up to 1.

$$f_j(z) = \frac{e^{z_j}}{\sum_{k=1}^{n} e^{z_k}}, \tag{5}$$

where z is an arbitrary vector with real values zj, j=1,. . . ,n, generated at the ith layer of the CNN and n is the size of the vector. The (i+1) st layer is called the softmax layer. The classification performed by the CNNs is accomplished at the final layer of the network. In particular, for a CNN which consists of i+1 layers, the softmax function is used to transform the real values generated by the ith CNN layer to possibilities [5]

- In a Keras layer, the input shape is generally the shape of the input data provided to the Keras model while training. The model cannot know the shape of the training data. The default input size for this model is 224x224.

- Pooling is primarily the process of further sub sampling of the feature map without losing information. Also, it prevents the over fitting of the model. After convoluting the image, its output is used as input, and it is sub-sampled to generate a single output. The pooling minimizes the number of parameters to compute while making the network insensitive to scale, shape, and size transformations. Max pooling is applied in the model which selects the image's brightest pixels [4] [9]

$$W2 = \frac{W1 - F}{S} + 1 \tag{6}$$

$$H2 = \frac{H1 - F}{S} + 1 \tag{7}$$

$$D2 = D1 \tag{8}$$

Pool Layer produces a volume [W2×H2×D2] where W2, H2, D2 are given by applying equations. [7]

- Dropout is a regularization technique which is used to reduce over fitting [3]. The Dropout layer is a mask that nullifies the contribution of some neurons towards the next layer and leaves unmodified all others. 0.25 dropout shows that 25 percent of input is randomly excluded from each cycle.

- Adam optimizer with a 0.25 dropout value shows a better result compared to SGD (Stochastic Gradient Descent) optimizer as it adaptively learns weights. In SGD, weights are updated incrementally after each epoch. Adam uses the following equation to update weights. [10] An accurate model is likely to be trained when the batch size is between 16 and 32 in CNN problems. When there is enough computing power, batch size can be reduced and get an even more accurate model. When the computing power is not enough, do a trade-off between efficiency and accuracy and choose a larger batch size. [9]

- The rescale=1. /255 will convert the pixels in the range [0,255] to the range [0,1]. Without scaling, the high pixel range images will have a large say to determine how to update weights. Also, the neural network has a higher chance of converging as it makes the coefficients in the range of [0,1] as opposed to [0,255] so helps the model process input faster.

- A flattening operation is applied to convert the data into a 1-dimensional array for inputting it into the next layer. And it is connected to the final classification model,

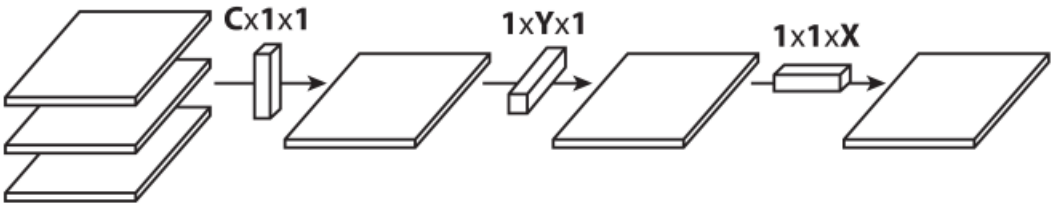

Figure 5: Flattening Operation

which is called a fully connected layer. [14]

- As the number of epochs increases, more times the weight is changed in the neural network and the model goes from underfitting to optimal to overfitting curve. An entire dataset is passed forward and backwards through the neural network once in each epoch.

- ModelCheckpoint call back is used in conjunction with training using model.fit() to save a model or weights at some interval, so the model or weights can be loaded later to continue the training from the state saved. When save_best_only=True, it only saves when the model is considered the **best** and the latest best model according to the quantity monitored will not be overwritten.

- Keras framework provides early stopping functionality which allows tracking a metric and stopping the training when it raises or falls below the metric value in its previous training epoch. As the research work is dealing with overfitting with the base model, it is required to stop the training once the validation loss begins to rise and not wait until 100 epochs. Thus, early stopping is used to monitor the validation metric and stop the training once the metric shows a raising trend as it is expected to minimize the validation loss. [12]

6 Implementation

- Convolution Layer 1: It takes images as input and applies convolution with 3*3 Filter Size and a total of 16 Filters.

- Convolution Layer 2: It takes Output of Max Pooling Layer 1 as input and applies convolution with 3*3 Filter Size and a total of 36 Filters.

- Max Pooling Layer 1: It takes the output of Convolution Layer 1 as input and applies Max Pooling with of Size 2*2 with stride 2*2 which reduces the image height and width to half.

- Convolution Layer 3: It takes the Output of Convolution Layer 2 as input and applies convolution with 3*3 Filter Size and a total of 64 Filters.

- Max Pooling Layer 2: It takes the output of Convolution Layer 3 as input and applies Max Pooling with Kernel of Size 2*2 with stride 2*2 which reduces the image height and width to half.

- Convolution Layer 4: It takes the Output of Max Pooling Layer 2 as input with 3*3 Filter Size and a total of 128 Filters.

- Max Pooling Layer 3: It takes the output of Convolution Layer 5 as input and applies Max Pooling with Kernel of Size 2*2 with stride 2*2 which reduces the image height and width to half.

- Dropout Layer: It takes the output of Max Pooling Layer 3 as input and randomly drops neurons with a dropout rate = 0.25.

- Flattening Layer: This layer transfigures the multidimensional tensor output from the dropout layer to a one-dimensional tensor.

- Fully Connected Layer 1: It takes the output of the flattening layer and returns the layer after applying ReLU to add non-linearity to it.
- Fully Connected Layer 2: This layer receives the output from the first fully connected layer and returns the layer without applying ReLU as this contains the probability for each class.
- Softmax Layer: It converts the score of each class into Probability Distribution. The softmax function is used as the last activation function of a neural network to normalize the output of a network to a probability distribution over predicted output classes.
- A top up is created to upload an image to the model and it can be used to re-run.
- Model Summary: Figure 6 explains the summary of the model. When the second convolution layer is added input is decreased by 2. That is from 142 to 140

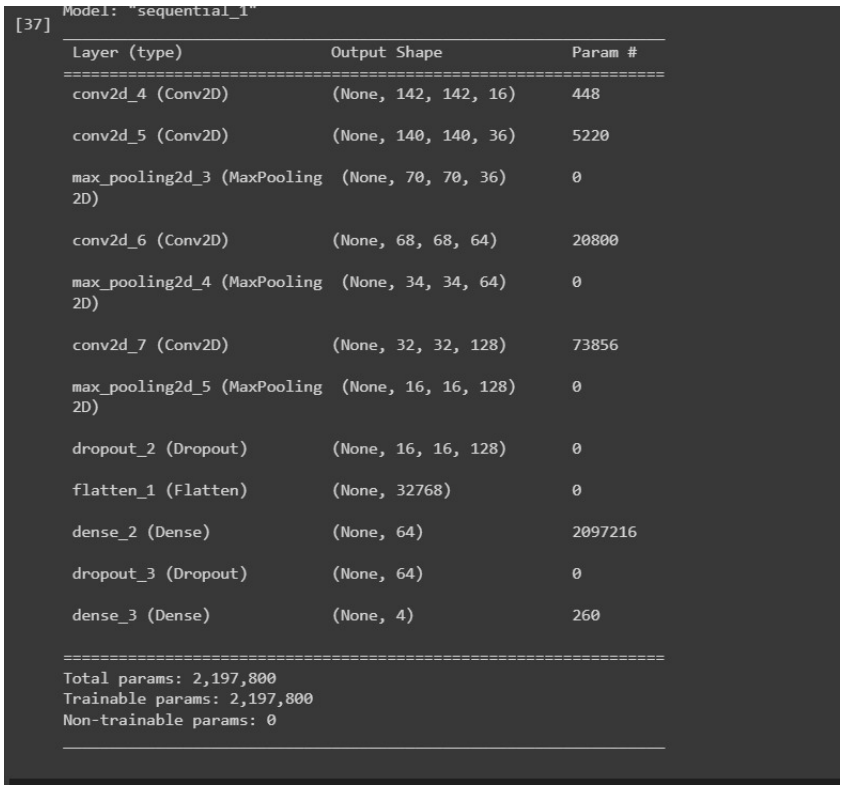

Figure 6: Model Summary

7 Results

Once the image is inserted using the top up created, Model is successfully able to differentiate the Bagel or BagelDog and Muffin or Chihuahua Dog. Based on different features, texture, shapes of the images, Model is able to identify the images like a human vision. From the

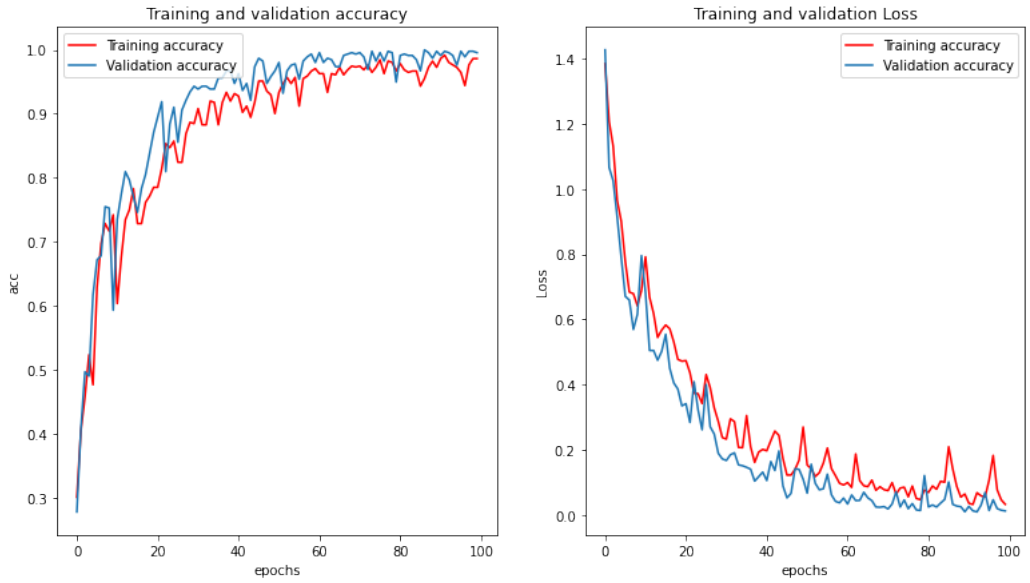

Figure 7: Accuracy and Loss of training and validation dataset for different numbers of epochs

figure 7, the accuracy of training and validation data can be seen to increase with the epochs number 1 to 100 and reached about 99 per cent. The second graph from figure 4 shows the loss value at each epoch for the training and validation data set. About 99 percent of accuracy is received from the model on the testing data set.

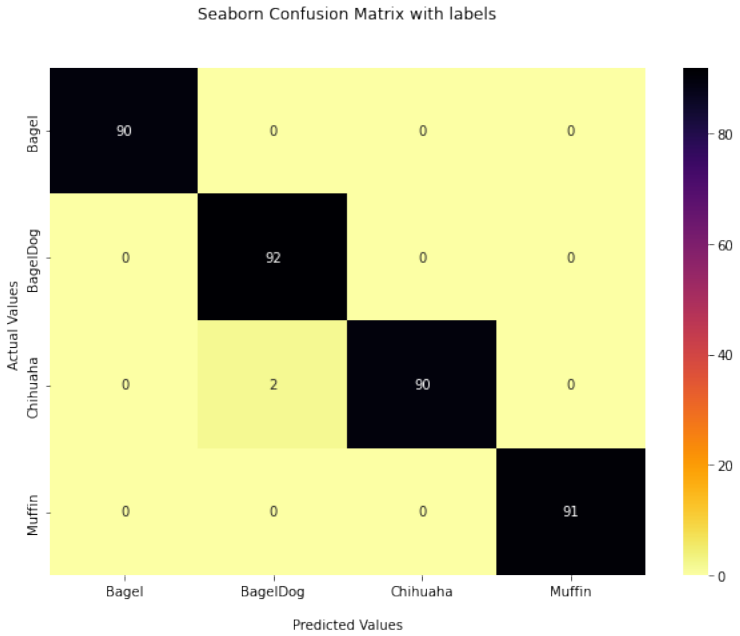

Figure 8: Confusion

Model performance can be validated in different ways.

Figure 8 explains the validation result for the model. It is calculated using one of the popular methods is using the confusion matrix. [2] Diagonal values of the matrix indicate correct predictions for each class, whereas other cell values indicate several wrong predictions. Chihuahua and BagelDog are misclassified. 2 images having Chihuahua are classified as BagelDog.

8 Limitations

- As the number of layers of a neural network increases, the features that can be extracted are more complex, and eventually, the convolution neural network uses it to make judgements about the features learned. Nonetheless, if a standard multi-layer perceptron is used, which means all layers are fully connected, CNN will soon become having difficulty in calculating because the dimension of the image is too high. [6]
- CNN fails to give the better model if the data set is too small to train the data. Hence most of the times, Data Augmentation is necessary.
- CNNs have a big loss of information in the pooling layer, reducing the spatial resolution. Therefore, CNNs will not be able to distinguish differences in postures and other facets.

9 Conclusion

The project shows the CNN model with improved performance for the Multi-class image classification. Here the images chosen are difficult to differentiate easily. The proposed CNN model also can be used efficiently for low-quality datasets. In this experiment, the multi-class dataset is divided for training, validation, and testing purposes. To achieve high accuracy multiple times the network's training is conducted with different combinations of convolution layers and hidden layers with techniques of batch normalization, dropout, activation, and max pooling have been employed in the hidden layers. The network training with the augmented image dataset increases the accuracy up to 99 percentage which describes the importance of data augmentation before feeding images into the model. The experiment shows batch normalization in the fully connected layer also increases the overall performance. Since the images like bagel vs BagelDog and muffin vs Chihuahua are rare, CNN cannot perform well and this can also be avoided with data augmentation. Loss of data can be avoided using appropriate dropout value.

References

[1] Ajit, A., Acharya, K., and Samanta, A. A review of convolutional neural networks. In *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)* (2020), IEEE, pp. 1–5.

[2] Chagas, P., Akiyama, R., Meiguins, A., Santos, C., Saraiva, F., Meiguins, B., and Morais, J. Evaluation of convolutional neural network architectures for chart image classification. In *2018 International Joint Conference on Neural Networks (IJCNN)* (2018), IEEE, pp. 1–8.

[3] Chauhan, R., Ghanshala, K. K., and Joshi, R. Convolutional neural network (cnn) for image detection and recognition. In *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)* (2018), IEEE, pp. 278–282.

[4] Islam, M. A. Reduced dataset neural network model for manuscript character recognition.

[5] Iwana, B. K., Kuroki, R., and Uchida, S. Explaining convolutional neural networks using softmax gradient layer-wise relevance propagation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (2019), IEEE, pp. 4176–4185.

[6] Jiang, X., Wang, Y., Liu, W., Li, S., and Liu, J. Capsnet, cnn, fcn: Comparative performance evaluation for image classification. *International Journal of Machine Learning and Computing 9*, 6 (2019), 840–848.

[7] Jmour, N., Zayen, S., and Abdelkrim, A. Convolutional neural networks for image classification. In *2018 international conference on advanced systems and electric technologies (IC_ ASET)* (2018), IEEE, pp. 397–402.

[8] Lee, Y. Image classification with artificial intelligence: Cats vs dogs. In *2021 2nd International Conference on Computing and Data Science (CDS)* (2021), IEEE, pp. 437–441.

[9] Lin, R. Analysis on the selection of the appropriate batch size in cnn neural network. In *2022 International Conference on Machine Learning and Knowledge Engineering (MLKE)* (2022), IEEE, pp. 106–109.

[10] Mehta, S., Paunwala, C., and Vaidya, B. Cnn based traffic sign classification using adam optimizer. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)* (2019), IEEE, pp. 1293–1298.

[11] Poojary, R., and Pai, A. Comparative study of model optimization techniques in fine-tuned cnn models. In *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)* (2019), IEEE, pp. 1–4.

[12] Prakash, S. S., and Visakha, K. Breast cancer malignancy prediction using deep learning neural networks. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)* (2020), pp. 88–92.

[13] Sultana, F., Sufian, A., and Dutta, P. Advancements in image classification using convolutional neural network. In *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)* (2018), pp. 122–129.

[14] Tripathi, S., and Kumar, R. Image classification using small convolutional neural network. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (2019), IEEE, pp. 483–487.

[15] Wang, Y., Li, Y., Song, Y., and Rong, X. The influence of the activation function in a convolution neural network model of facial expression recognition. *Applied Sciences 10*, 5 (2020), 1897.