Object Classification

1. ABSTRACT

The objective of this project is to compare different models capable of classifying general objects that we observe in our day-to-day life. The dataset used is CIFAR-10 dataset. We implement a basic Convolutional Neural Network architecture followed by GoogLeNet architecture which contains the Inception module. We develop the CNN architecture from the ground up and use transfer learning approach to implement the GoogLeNet architecture. The latter is a pretrained model which is imported and trained on the CIFAR-10 dataset for the object classification task. Post training, the developed models were evaluated on the validation set and test set with the metrics of accuracy and precision. It could be noted that the CNN model achieved an accuracy and precision of 84 % and 76 % on the test set, whereas the GoogLeNet model performed better with an 89.13 % accuracy.

2. INTRODUCTION

The amount of data being generated was projected to reach about 65 zettabytes in the year 2020, and over the next 5 years, it was forecasted to grow over 180 zettabytes. One of the causes of such rapid increase was the COVID-19 pandemic where increased people worked and learned from home and chose home entertainment options quite frequently. The increase in data has propelled researchers to develop increasingly sophisticated programs that can support us humans in our day-to-day activities. Some of the common examples of this are in the field of security, recommendations systems for online purchasing, fraud detection, development of self-driving cars for commuting, etc. One of the core developments is the task of object classification wherein the model identifies which object is presenting an image. This is used in a variety of applications like self-driving cars, security systems, retail analytics, etc.

Object classification is a computer vision task which simply identifies and classifies objects present in an image. The main hurdle in such tasks is that objects in the image can be of several types such as humans, automobiles, animals, etc. Each of the objects differs in their size, shapes, colors, orientation, and location in the image. The other factors (external) that affect the task of object classification are the resolution of the image captured, the device used to capture the image, etc. There are mainly 2 approaches for object classification being used actively by professionals and researchers in the field, one being the traditional methods which utilize hand-crafted features to represent images. Traditional methods are often used to classify objects in simple images. The other approach applies Deep learning models which utilize Convolutional neural networks to automatically learn features from images. The most common procedure uses supervised learning for training a deep learning for such a task.

For this project, we implement a simple Convolutional Neural network, and the GoogLeNet (Inception) architecture for object classification. The models are trained and evaluated on the training and validation set of the CIFAR10 dataset, respectively. The choice of models was dependent on factors such as relative performance based on the computational power available in comparison to the state-of-art

architectures such as YOLOv8, Vision Transformers, OpenAI CLIP, etc. The CNN model has been developed from the ground up and the GoogLeNet model is a pretrained model which has been trained on the ImageNet dataset. The pretrained model is then fine-tuned on the CIFAR10 dataset for the purpose of classifying 10 objects/classes in it. To enable model training, the dataset is loaded and split into training, validation, and test sets. A few pre-processing steps like converting image arrays to tensors and normalization of the dataset were also incorporated. Following preprocessing, the model trained using the cross-entropy loss function and various iterations of model training were performed using different optimizers, and hyperparameters including weight decay, momentum, and learning rate.

3. BACKGROUND

The CIFAR-10 dataset is a popular benchmark dataset for machine learning algorithms, especially for image classification. It is a relatively small dataset, which makes it easy to work with and to train models on. However, it is also a challenging dataset, as the images are of high quality and the classes are relatively similar. An active leaderboard for the image classification on CIFAR-10 is also maintained online. The prominent models that have the best performance till date are ASF-former-B model with an accuracy of 98.8%, the second is ASF-former-B model with an accuracy of 98.7%, followed by IM-Loss (ResNet) model with an accuracy of 95.49% on the test set. There have been a lot of developments in the deep learning field in terms of architecture developed particularly for image classification. Some of the prominent models include ResNet, AlexNet, Inception_v4, YOLO, etc.

Although CIFAR-10 dataset is a popular benchmark for object classification, there are a few limitations. One of the drawbacks of the CIFAR-10 dataset is that it is small. This can make it difficult to train models that are able to generalize well to new data. Additionally, the classes in the CIFAR-10 dataset are similar, which can make it difficult for models to distinguish between them. Another drawback of the CIFAR-10 dataset is that it is not truly diverse. The images in the dataset are all objects that are commonly found in the real world, but they do not represent the full range of possible images. This can make it difficult for models trained on the CIFAR-10 dataset to generalize to new images that are outside of the dataset. Despite its drawbacks, the CIFAR-10 dataset is a valuable resource for machine learning researchers.

Many latest developments in image classification have been seen. Some of the prominent developments are Vision Transformers (ViTs). These are a new type of deep learning model based on the Transformer architecture, originally developed for natural language processing. ViTs have been shown to achieve state-of-the-art results on image classification tasks, and they are much more efficient than traditional convolutional neural networks (CNNs). Attention mechanisms have been integrated into Transformer-based models to improve their performance on image classification tasks. Hybrid models that combine CNNs with Transformer-based models have also been shown to achieve state-of-the-art results on image classification tasks. Cross-modal learning is a technique where a model is trained on multiple modalities, such as images and text, to learn joint representations. This can improve the model's performance on image classification tasks, as the model can learn from both the visual and textual features of the image.

4. APPROACH

4.1. Convolutional Neural Networks

CNNs are a type of artificial neural network that is specifically designed for image processing. Because they can learn characteristics from images in a hierarchical way, CNNs are exceptionally good at tasks like image segmentation, object identification, and classification. Convolutional filters are applied to the input image in a series to make CNNs function. Small arrays of weights called convolutional filters are used to extract features from images. Repeated application of the filters at various points in the image enables CNN to learn characteristics at various scales. A max pooling layer is often applied by the CNN after the convolutional filters have been applied. The max pooling layer minimizes the size of the convolutional filters' output, which aids in lowering the network's computational complexity. Next, a string of completely connected layers receives the output from the max pooling layer. The input image is categorized by the fully connected layers.

The CNN that we have implemented in this paper 4 convolutional layers with 3 x 3 filters and Rectified Linear Unit (ReLU) activation function. Two MaxPooling layers and 2 Dropout layers were also implemented before flattening and feeding to the dense layers. We used 2 Dense layers with ReLU activation before feeding to the output layer with Softmax activation function. Categorical Cross-entropy loss function was employed for this model with a learning rate of 0.0001 achieving 76% precision and 0.758 F1 score.

4.2. GoogLeNet (Inception)

In 2014, researchers at Google AI created the convolutional neural network (CNN) architecture known as GoogleNet. It was a breakthrough architecture that they used to win the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) featured several ground-breaking concepts. The usage of inception modules is one of GoogleNet's major innovations. Convolutional filters can be combined at various scales using inception modules. This makes GoogleNet more resilient to changes in object size and pose by enabling it to learn features at many scales. The usage of dropout is yet another significant breakthrough of GoogleNet. Dropout is a technique that, while a neural network is being trained, randomly removes some of its nodes. This helps avoid overfitting, which can happen when a neural network is trained on a large dataset and is a concern. Although GoogleNet has a fairly complicated architecture, it also works quite well. On the ILSVRC 2014 challenge, GoogleNet earned a top-5 error rate of 6.67%, which was a substantial improvement over the prior state-of-the-art. Several tasks, such as image classification, object identification, and picture segmentation, have been carried out using GoogleNet. The development of computer vision has been significantly impacted by the potent architecture known as GoogleNet.

5. RESULTS

5.1. *Dataset*

We implement the models on the CIFAR-10 dataset. The CIFAR-10 dataset is a collection of 60,000 32x32 color images in 10 classes, with 6,000 images per class. The classes are Airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The data was collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. They started with a dataset of 80 million tiny images, which were downloaded from the internet. They then manually selected 60,000 images from this dataset to create the CIFAR-10 dataset. The images in the CIFAR-10 dataset were then labeled by hand and the labeling was done by a team of students at the University of Toronto. The images were then resized to 32x32 pixels and converted to RGB color space. The CIFAR-10 dataset is a valuable resource for machine learning researchers because it is a challenging dataset that is small and easy to work with.

The dataset was segregated into the following splits: Train Set: 45000 images, Validation Set: 5000 images, Test Set: 10000 images. Each set was randomly shuffled so that bias can be reduced.

5.2. Experiments & Performance Evaluation

To evaluate the model's metrics of accuracy and precision were used. Accuracy is the proportion of predictions that the model makes correctly. Precision is the proportion of positive predictions that the model makes correctly. This is computed based on the correctness of classification of the images by the model.

Experiment Details:

CNN:

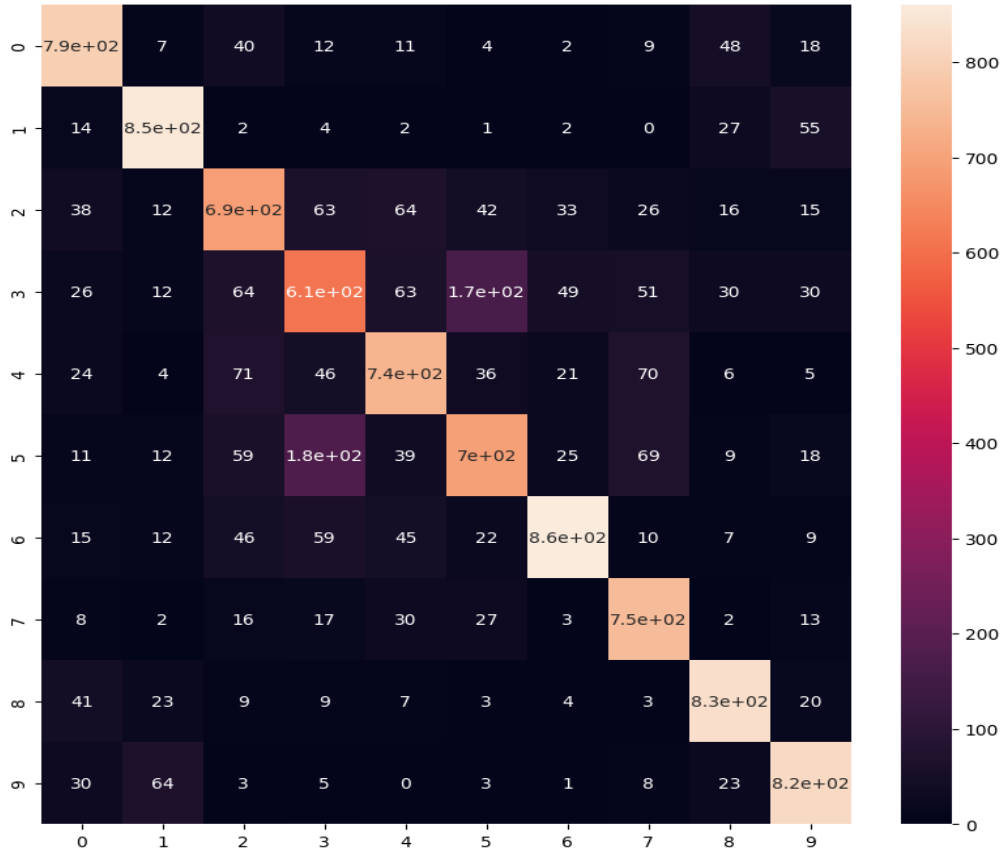
We first use the basic CNN architecture designed from the ground up. We use this as we would like to first set a baseline model for the image classification task on CIFAR10 dataset. The architecture used are summarized are follows:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 30, 30, 32)        896

 conv2d_1 (Conv2D)           (None, 28, 28, 32)        9248

 max_pooling2d (MaxPooling2  (None, 14, 14, 32)        0
 D)

 dropout (Dropout)           (None, 14, 14, 32)        0

 conv2d_2 (Conv2D)           (None, 12, 12, 64)        18496

 conv2d_3 (Conv2D)           (None, 10, 10, 64)        36928

 max_pooling2d_1 (MaxPoolin  (None, 5, 5, 64)          0
 g2D)

 dropout_1 (Dropout)         (None, 5, 5, 64)          0

 flatten (Flatten)           (None, 1600)              0

 dense (Dense)               (None, 1024)              1639424

 dropout_2 (Dropout)         (None, 1024)              0

 dense_1 (Dense)             (None, 1024)              1049600

 dense_2 (Dense)             (None, 10)                10250

=================================================================
Total params: 2764842 (10.55 MB)
Trainable params: 2764842 (10.55 MB)
Non-trainable params: 0 (0.00 Byte)
```

We specifically use drop-out layers in between the layers to prevent overfitting. Overfitting occurs when a model learns the training data too well and is unable to generalize to new data. Dropout layers randomly drop out (or set to zero) a certain percentage of neurons in a layer during training. This forces the model to learn to rely on more than just a few neurons, which can help to prevent overfitting.

Post training the model was evaluated based on the metrics of precision and accuracy, and a confusion matrix was plotted to indicate the performance of the multi-class classification on CIFAR10 dataset.

	Accuracy	**Loss**
Training	0.8445	0.4424
Validation	0.7641	0.7080



GoogLeNet:

We use the pretrained model of GoogleNet which uses the Inception module. The model was loaded using PyTorch's torchvision module. We train the model on the complete training set and evaluate on the validation set.

Two variations of the GoogLeNet were used for the model training part:

The first model was trained with the hyparameters of gamma = 0.1, step size = 7, learning rate of 0.001 and a momentum of 0.9. The model was trained on 45 epochs and took 75 minutes to train. The final accuracy and loss of the model 1 were found out to be:

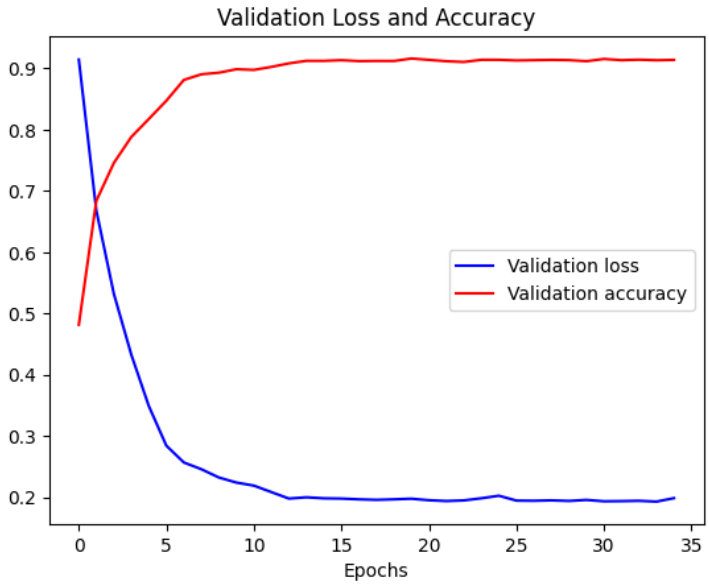
	Accuracy	**Loss**
Training	0.8728	0.3698
Validation	0.7280	0.8303

The second model was trained with the hyparameters of gamma = 0.1, step size = 7, learning rate of 0.0001 and a momentum of 0.9 and with the Nesterov momentum as True. This was trained for 35 epochs and took a total time of 60 minutes.

	Accuracy	**Loss**
Training	0.9140	0.2577
Validation	0.9470	0.1985

The model 2 was seen to have the training plots for accuracy:

The model 2 was seen to have the validation plots for accuracy:



With the promising performance of the second model, we selected the model 2 as our best performing model. The model selection process concluded at this stage and model 2 was implemented on the test set. The accuracy was found to be 89.13%.

Some of the predictions on the test set could be observed with good accuracy:

Results Analysis:

The best model for the task of Object classification on CIFAR10 dataset was found to be GoogLeNet trained with Nesterov Momentum and weight decay parameters. The accuracy of the model was found to be 94.7% on the validation set as compared to the accuracy of 73% using CNNs. The model results are compared through running and evaluating the models on the same number of training samples. The performance would further increase considering training the models on enormous number of training samples and increasing the Hyperparameter combinations. The GoogLeNet model which contains chunks of Inception blocks are designed to extract features from images at different scales, this can improve the performance of the CNN on image classification tasks.

6. **DISCUSSION**

Through this work we could see that GoogLeNet outperformed CNN by a considerable margin. While CNN architecture achieved training accuracy of 82% its precision and F1 scores are 0.77 and 0.76, respectively. On the other hand, GoogLeNet achieved over 89.19% training accuracy on test data.

The finding of this work is what we hoped for as GoogleNet is an immensely powerful CNN architecture that has been shown to achieve state-of-the-art results on a variety of image classification tasks. It is also relatively efficient, which makes it suitable for real-world applications. However, GoogLeNet is a complex architecture, which can make it difficult to train and deploy. It can also be less accurate than some other CNN architectures on small datasets.

Future work could include research using other deep learning architectures like YOLO, MobileNet which employ single shot learning. They are computationally efficient and can be implemented in real time and on edge devices as well. We also plan to employ different datasets to draw conclusions on how GoogLeNet performs against CNN architectures. To generalize, the model's future work can also include image augmentation techniques.

7. CONCLUSION

The goal of this project was to develop an object classification model by utilizing advanced architectures like Inception. The GoogLeNet model was trained was on the CIFAR10 dataset that comprised a total of 60000 labelled images. The training was done on 45000 labelled images in a supervised learning way. The training set was preprocessed where image normalization and transformation to tensors was performed. The trained models evaluated on the validation set based on the model accuracy metric. It could be clearly seen that advanced architecture of GoogLeNet outperformed the CNN architecture by at least 15%. This can be attributed to the longer training time of GoogLeNet model on the ImageNet dataset where it could already learn the different weights pertaining to existing images. This allowed the model to capture more features in the images as compared to CNN, leading to a better classification score.

Overall, our project investigated the importance of using inception modules for the task of object classification. We look at the importance of selecting the appropriate model based on model performance, and available resources. These models have the power to alter the ways we interact with the available image data and have numerous applications that can aid us in our day-to-day activities.

8. REFERENCES

[1]. Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

[2]. https://www.statista.com/statistics/871513/worldwide-data-created/

[3]. Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009): 7.

[4]. Imambi, Sagar, Kolla Bhanu Prakash, and G. R. Kanagachidambaresan. "PyTorch." *Programming with TensorFlow: Solution for Edge Computing Applications* (2021): 87-104.

[5]. Ho-Phuoc, Tien. "CIFAR10 to compare visual recognition performance between deep neural networks and humans." *arXiv preprint arXiv:1811.07270* (2018).

*Each team member in our group has contributed equally to the success of this project.