Report: Analysis of Placement Data (BrainDead, Revelation 2k23)

Team Members: Dibyarup Dutta, Anuvab Sen, Mritunjay Haldar, Abhiroop Mukherjee, Aritra Bandyopadhyay

University: Indian Institute of Engineering Science and Technology, Shibpur

Overview

Challenge Description

The main goal of this project is to analyze the factors affecting the factors that affect the placement and salary of the students. The dataset contains information about the placement records of students in a MBA college.

Dataset Description

The dataset includes variables such as secondary and higher secondary school percentages, degree specializations, work experience, and the salary offered to the students.

Relevant Links:

1. **Data Analysis Link**
   https://www.kaggle.com/code/anuvabsen123/thedatadorks/notebook

Dataset Exploration

- **Data Exploration on the Entire Dataset:** We first used pairplot and regression analysis on the entire dataset to derive insight of the entire dataset

- **Cluster Exploration based on Placement Status:** We divided the dataset into two parts based on placement status and derived insights from the two datasets

Cluster Exploration based on Specialization: We then divided the dataset into two parts based on specialization and then derived insights from the two divided datasets

Data Analysis Methods

We used various data analysis methods, including:

Pairplot:
Pairplot visualizes given data to find the relationship between them where the variables can be continuous or categorical.

Exploratory Data Analysis:
Various histograms were plotted to find out trends in Data so that these trends can be analyzed further

Correlation Matrix:
Linear Bivariate analysis method to find out correlations between various attributes.

Pie Charts:
Pie Charts were plotted that show trends among placed and non-placed students for various attributes

Bar Graphs:
Bar Graphs were plotted that helped in finding trends in min, max, median and average salaries and how they relate to various attributes

Box Plots:
Box Plots included show the capture trends and show information about which percentile of a certain class gets more salary, it is a standardized way of displaying the dataset based on the five-number summary: the minimum, the maximum, the sample median, and the first and third quartiles

Results of Analysis

Observation: Salaries distribution are on the lower side of spectrum (Expected):

Analysis:

The salary range of the students in the dataset is between 200,000 and 940,000, with a mean salary of 288,655. The median salary is 265,000, which indicates that the salary distribution is slightly skewed towards the lower end of the spectrum.

This observation is in line with the expectations, as fresh MBA graduates usually start with lower salaries and gradually progress in their careers.

```
max          940000.000000
min          200000.000000
mean         288655.405405
median       265000.000000
Name: salary, dtype: float64
```

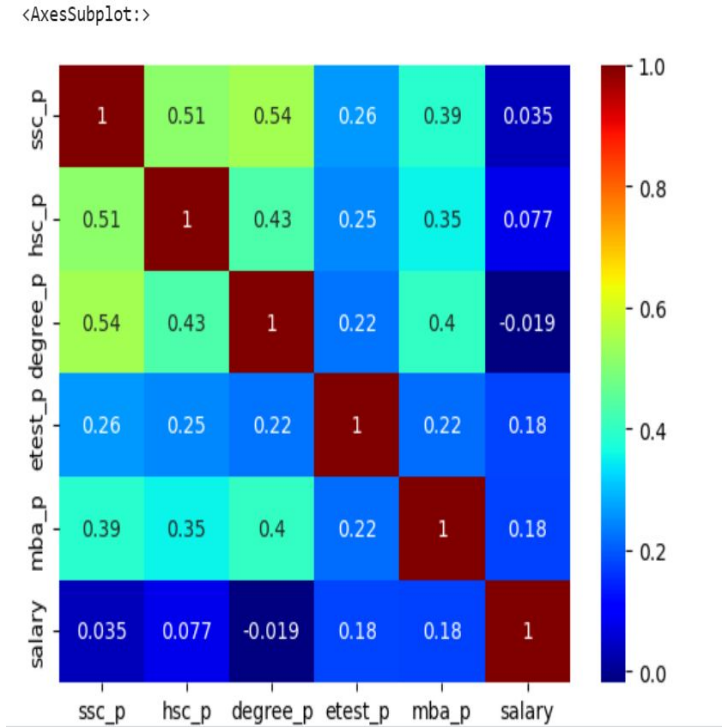
Observation: Correlation exists between degree and high school percentages

The dataset analysis revealed a moderate level of correlation between the high school and degree percentages of the students, indicating that students who performed well in high school tend to perform well in their degree programs (undergraduate as well as specialization).

Analysis:

However, the correlation coefficient is low, suggesting that factors such as work experience, specialization, and performance in the interview rounds are more important in determining the placement and salary of the students.

The salary is only slightly correlated to the MBA percentage as well as the employability test score, but no other inference for salary can be extracted from the data, perhaps there is some non-linear relationship that the correlation matrix cannot capture.

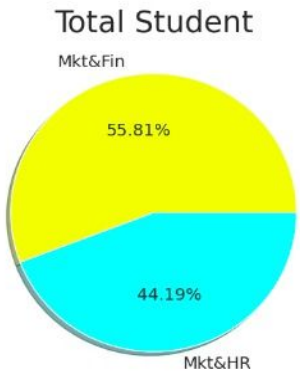


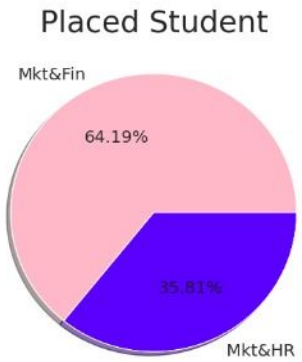
Observation: Placement Disparities Among MBA Specializations

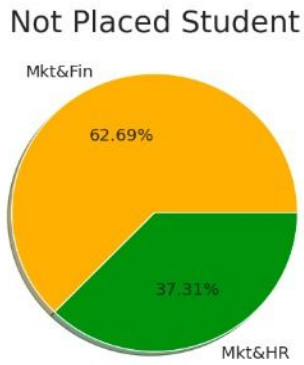
The analysis of the dataset revealed that students with specializations in Marketing and Finance had a higher number of placed students relative to their representation in the total student population.

Analysis: This suggests that these specializations are in high demand in the industry.
On the other hand, students with specializations in Marketing and HR had a lower number of placed students compared to their representation in the total student population, indicating that these specializations may not be as much in demand in the industry.

The findings suggest that students with specializations in Marketing and Finance may have a competitive edge in the job market compared to those with specializations in Marketing and HR. make it more concise





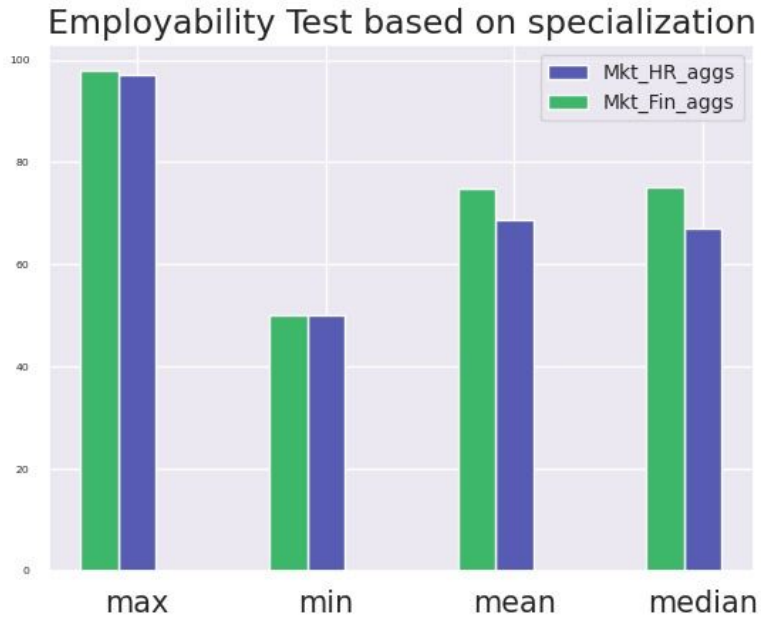


Observation: Employability Test Percentage based degree Specialization

Observation 1: The etest_p (Employability Test percentage) of Marketing and Finance students is higher compared to Marketing and HR students.

Observation 2: Marketing and Finance students received better job offers in general compared to Marketing and HR students.

Analysis: Based on the given observations, we can analyze that Marketing and Finance students have a better employability rate and also received better job offers compared to Marketing and HR students.

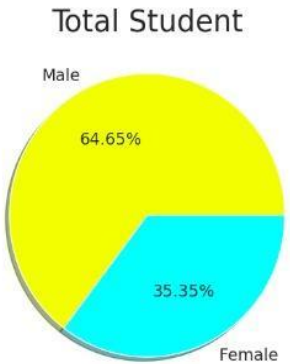


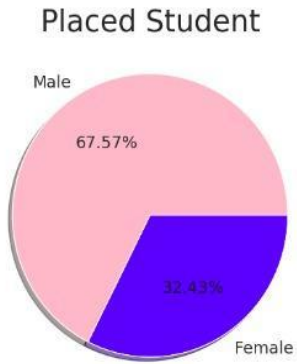
Observation: Analysis of Placements based on Gender and Total Population Representation

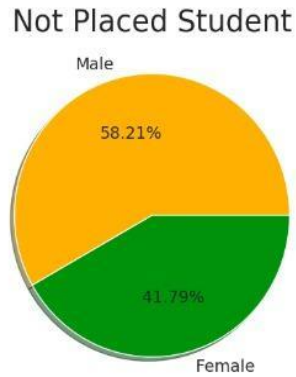
The placement rate is higher for students who have a higher representation in the total population, while female students have a lower placement rate relative to their representation in the total population.

Analysis: From the above pie charts we can conclude the following points-

- Students who have have a higher number of placed students in the aspect of their representation in the total population.
- Female Students have a lower number of placed students in the aspects of their representation in the total population. Their max salary is also 44.61% lower than Male.







Observations: Salary Analysis based on degree specializations

The mean, min, and median salaries were approximately the same for both Marketing and Finance students as well as Marketing and HR students,

However the maximum salary for Marketing and Finance students is significantly higher than Marketing and HR students.

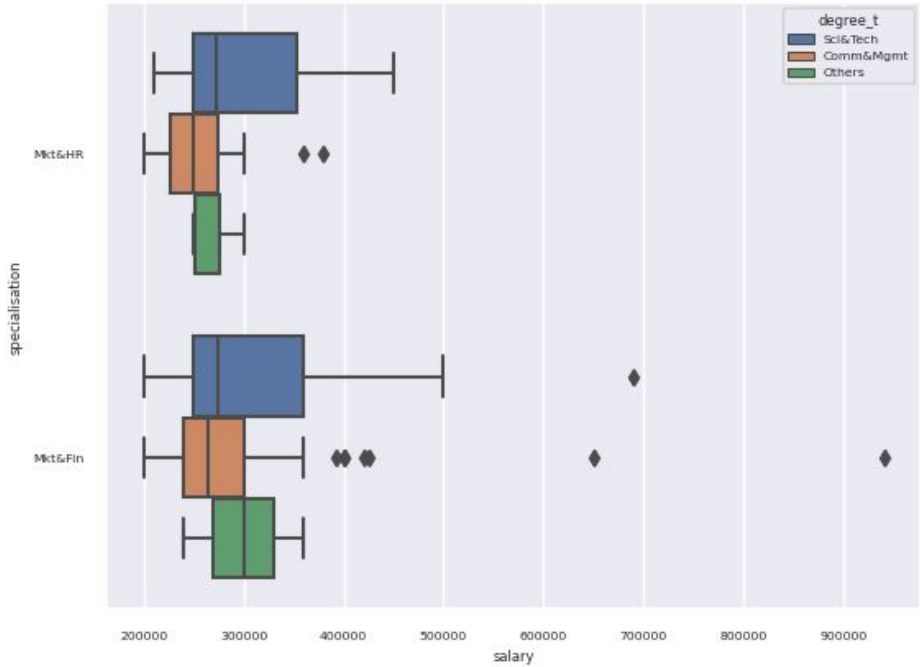
The significantly high outliers are responsible for the slight increase in mean salary for the Marketing and Finance students.

Observations: Salary Analysis based on undergraduate specializations

It is evident from the box plots that students with a specialization in Science and Technology in their undergraduate studies had a higher overall salary compared to students who pursued other degrees

Analysis:

One possible explanation for this correlation could be that Science and Technology degrees provide individuals with specialized skills and knowledge that are in high demand in the job market.



Dataset Analysis Link

Team Name: TheDataDorks
Link: https://www.kaggle.com/anuvabsen123/thedatadorks

Revelation 23

TheDataDorks

BRAINDEAD FINAL REPORT

Cartoon Emotion Recognition

Presented By Aritra Banerjee
Anuvab Sen
Abhiroop Mukherjee
Mritunjoy Halder
Dibyarup Dutta

THIS IS THE FINAL PRESENTATION OF THE EVENT BRAINDEAD, REVELATION.

CONTENT

Problem Statement

SCOPE OF STUDY

In the age of artificial intelligence and the internet of things, emotion recognition is essential.

It has enormous potential for behavioural modelling, robotics, healthcare, bio-metric security, and human-computer interface.

The future innovation in emotion recognition will allow machines to understand how people feel, which is the first step for them to fulfil our needs.

RELEVANCE OF STUDY

Cartoon emotion recognition is a challenging task for artificial intelligence systems, as the cartoon characters' expressions are often exaggerated and not realistic.
Advancements in this area can contribute to the development of more advanced artificial intelligence systems that can recognise emotions in real-world scenarios.

RESEARCH QUESTIONS

"What is the accuracy of deep learning algorithms in recognizing emotions in cartoon characters, and how does this accuracy compare to human performance?"
This research question could be further refined by specifying the types of emotions and the specific cartoon dataset used in the study. Additionally, the research could explore the factors that contribute to the accuracy of the deep learning algorithms and investigate potential methods for improving their performance.

DATASET IMAGE GENERATION

Content

Image Generation using AvatarGAN

AvatarGAN is a type of generative adversarial network (GAN) that can generate cartoon images using deep learning. It is based on the idea of training two neural networks: a generator that creates fake cartoon images from random noise, and a discriminator that tries to distinguish real cartoon images from the fake ones. The generator and the discriminator compete with each other to improve their performance. AvatarGAN can create cartoon images that do not belong to any human face, but resemble bitmoji or avatars

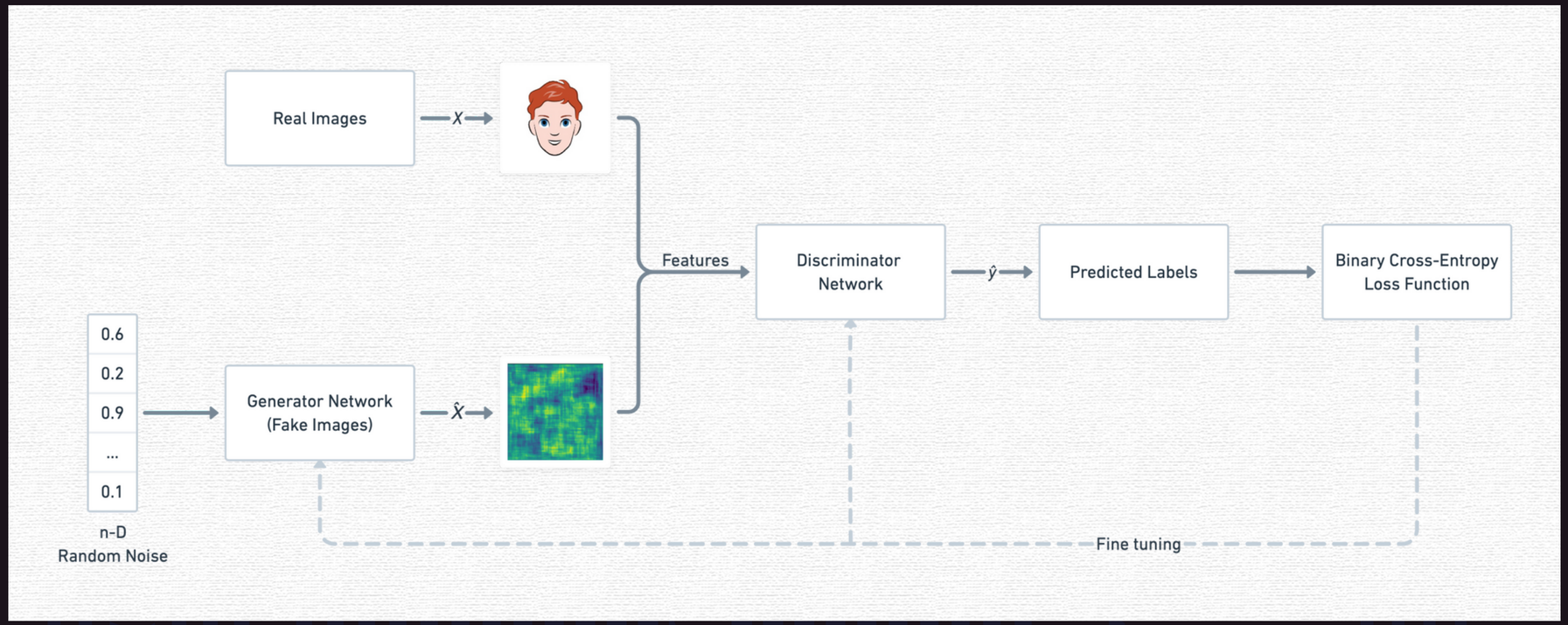


Image Generation using AvatarGAN

We found that the generated images are not up to our satisfaction so we decided to ignore GAN based models altogether. The image below are the example of generated image after 10 epochs.

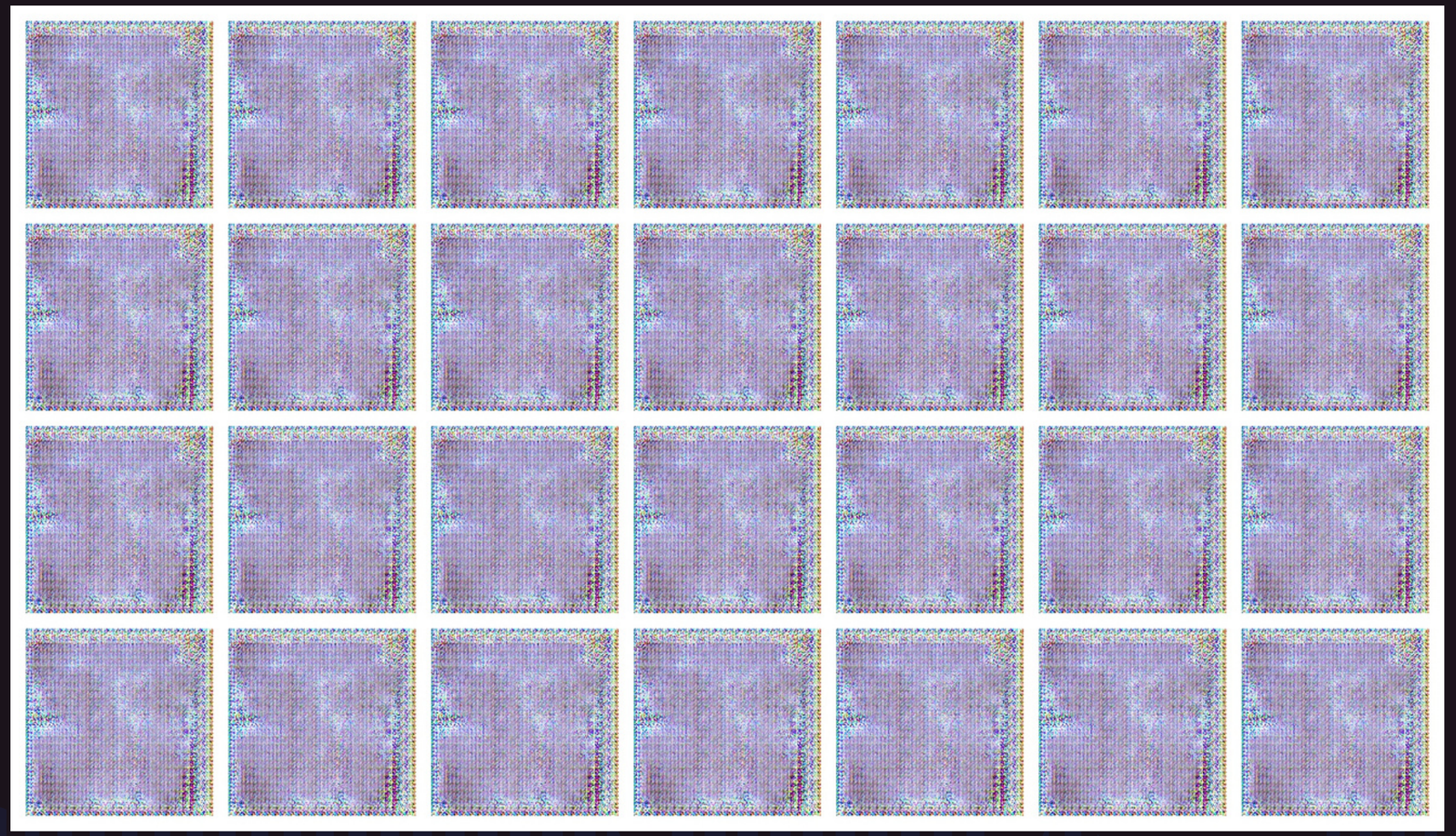


Image Generation using Stable Diffusion

Stable diffusion is a type of deep learning model that can generate realistic images from text descriptions. It uses a latent diffusion process, which gradually transforms random noise into an image that matches the text input. Stable diffusion can also perform other tasks such as inpainting, outpainting, and image-to-image translation guided by text.

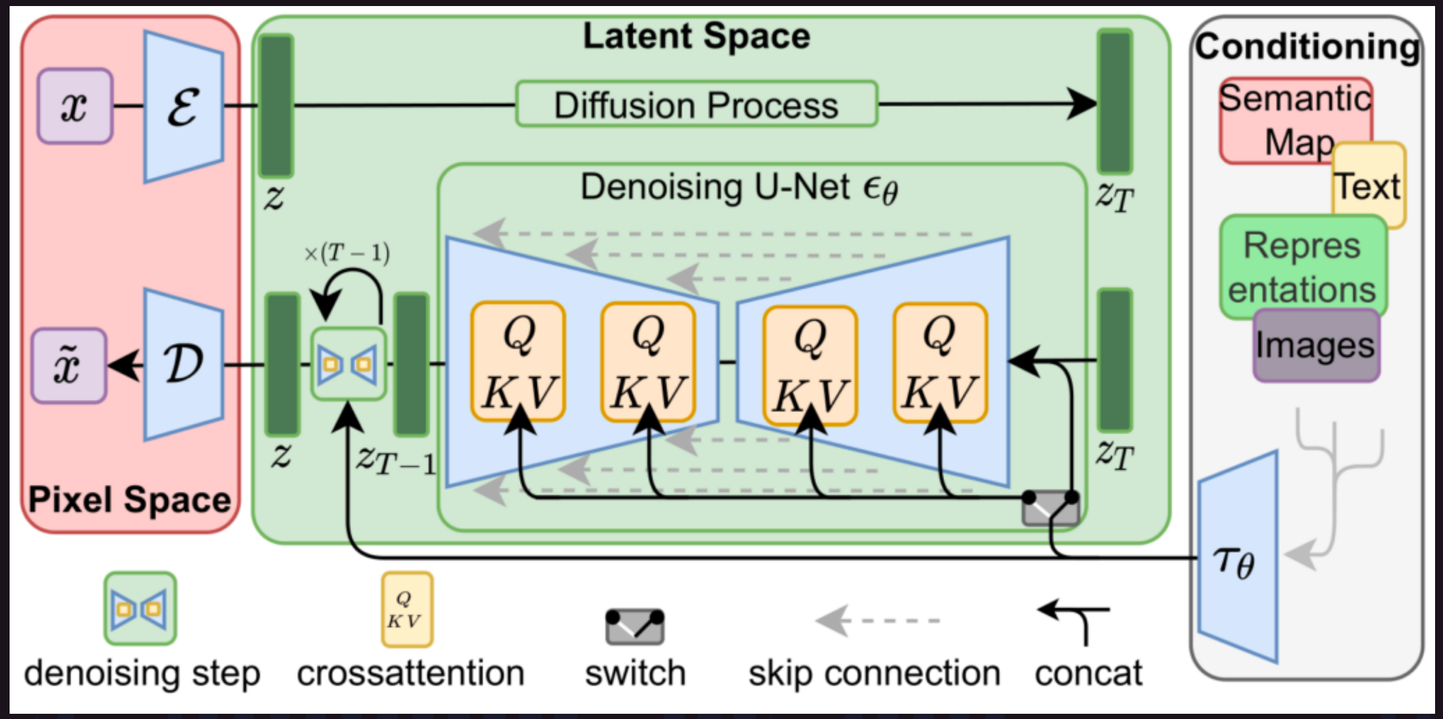


Image Generation using Stable Diffusion

By using Stable Diffusion to generate images from the given dataset, we received satisfactory results.

1. Stable Diffusion Image Variation Pipeline
2. Data Cleaning
3. Data Augmentation

Data Cleaning of Generated Images

Stable Diffusion is not perfect, it had a tendency to generate images that were not according to the required emotion. The model also occasionally created NSFW images. We used the in-built NSFW classifier on the stable diffusion model to ignore any generated NSFW images. Then we manually cleaned generated images to make sure that the generated images described the correct emotion.



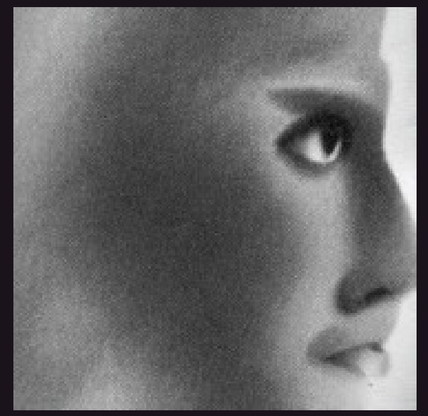



Image Systhesis using Image Augmentation

We augmented the generated images using rotation and horizontal flip as they do not change the target emotion. The generated, augmented and the given images combined forms our training data.




Image Systhesis using Image Augmentation

The model initially consists of a data augmentation block. The pipeline consists of several layers, which are executed sequentially. The first layer, Normalization, normalizes the pixel values of the input image to have zero mean and unit variance. The second layer, Resizing, resizes the input image to a fixed size specified by image_size. The third layer, RandomFlip, randomly flips the input image horizontally with a probability of 0.5. The fourth layer, RandomRotation, randomly rotates the input image by a small angle specified by factor. The fifth layer, RandomZoom, randomly zooms into the input image with a height and width factor specified by height_factor and width_factor, respectively. This data augmentation pipeline is useful in training deep learning models on image datasets, as it can help increase the amount of data available for training, improve the generalization of the model, and prevent overfitting

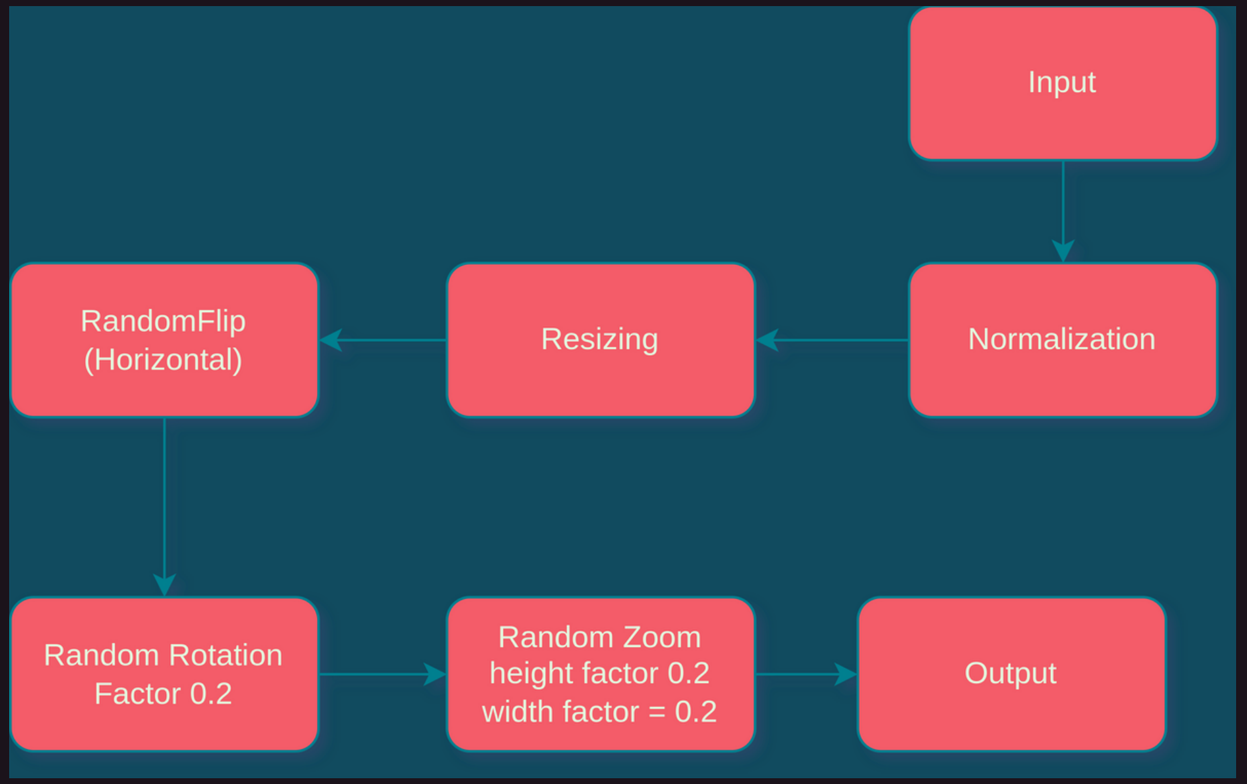
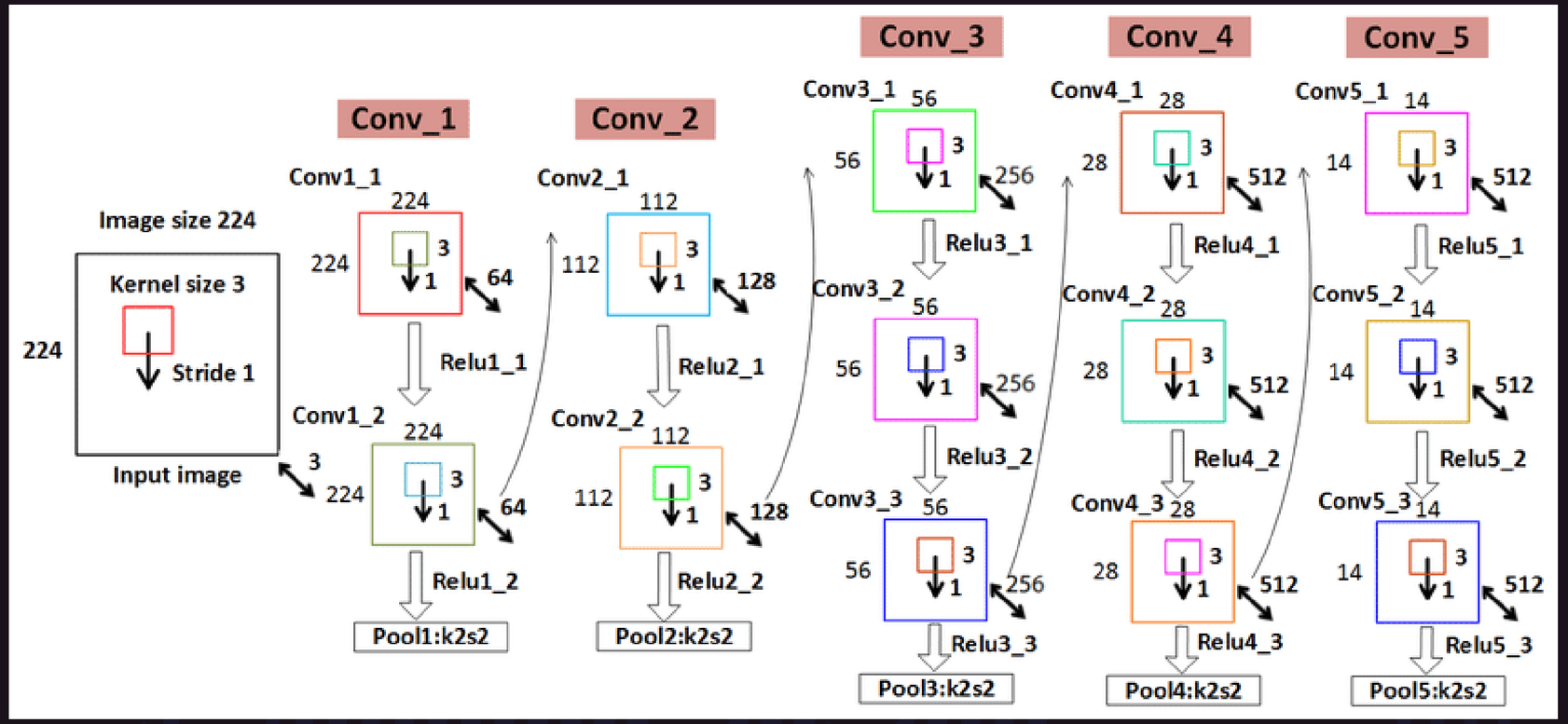


IMAGE CLASSIFICATION

Content

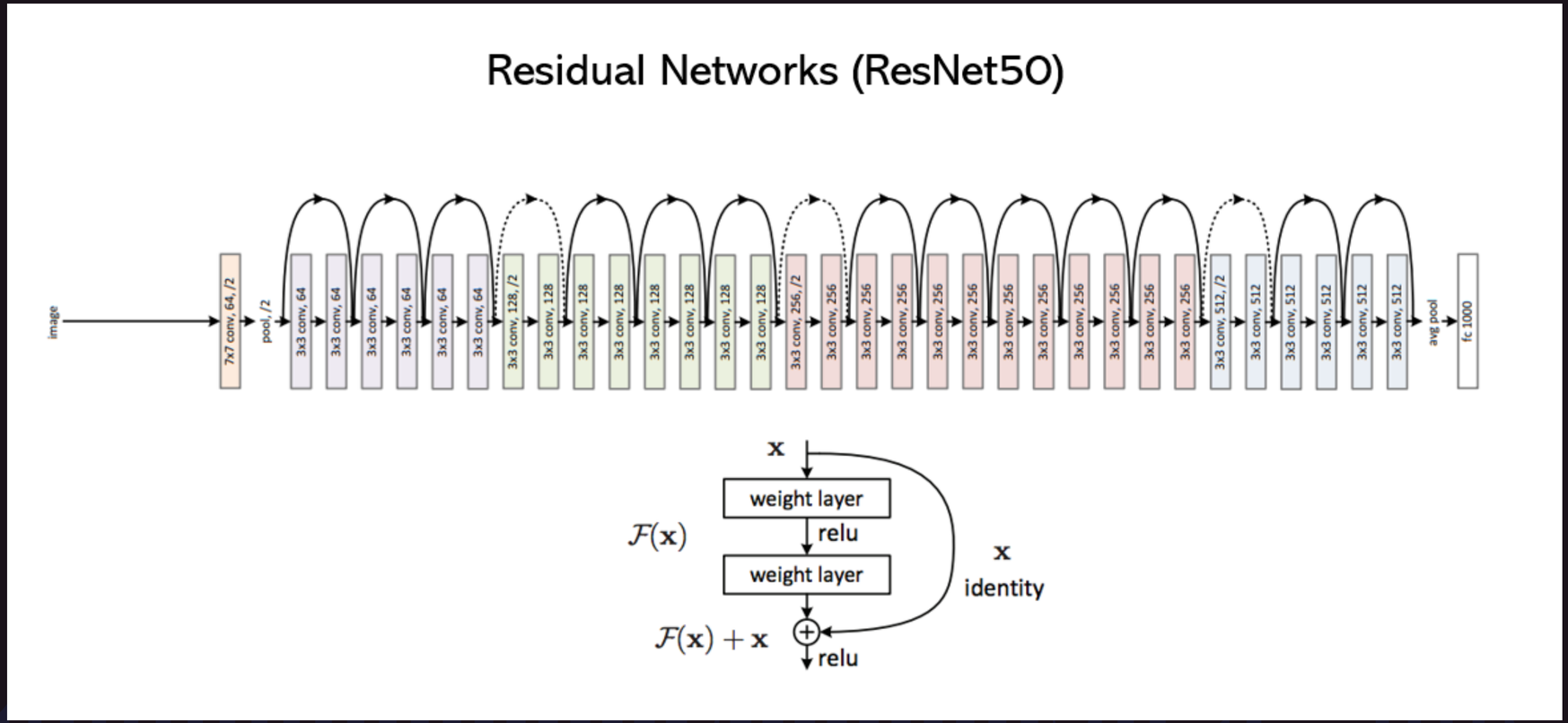
VGG16

A convolutional neural network with 16 layers is called VGG-16. The ImageNet database contains a pretrained version of the network that has been trained on more than a million photos. The pretrained network can categorise photos into 1000 different item categories, including several animals, a keyboard, a mouse, and a pencil.



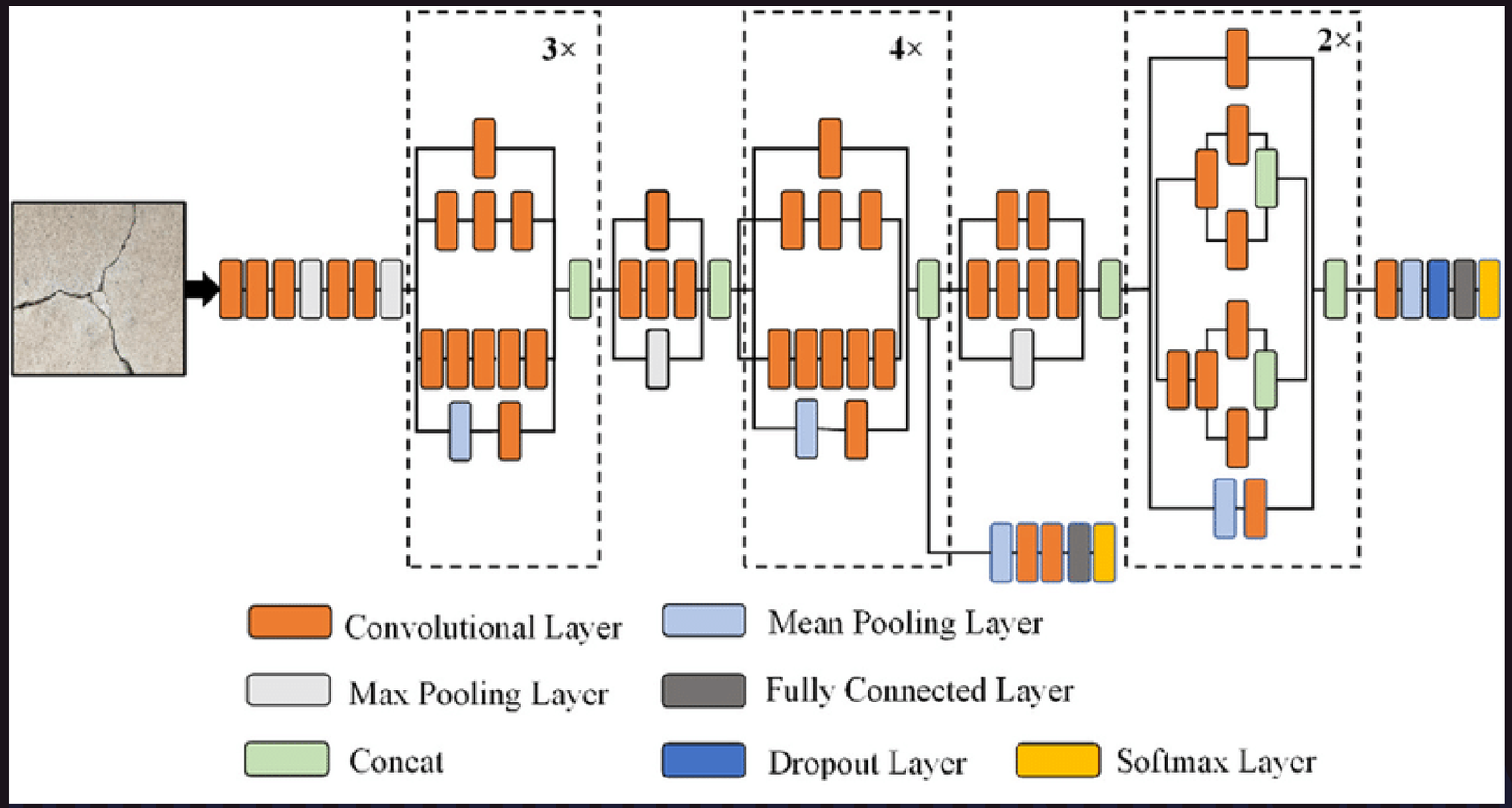
ResNet50

ResNet50 is a residual network that has 50 layers that uses skip connections or shortcuts to jump over some layers. This helps to avoid the problem of vanishing gradients and enables the network to learn from both shallow and deep features. ResNet50 can classify images into 1000 categories, such as animals, objects, and scenes. It can also be used as a backbone model for other tasks, such as object detection and segmentation.



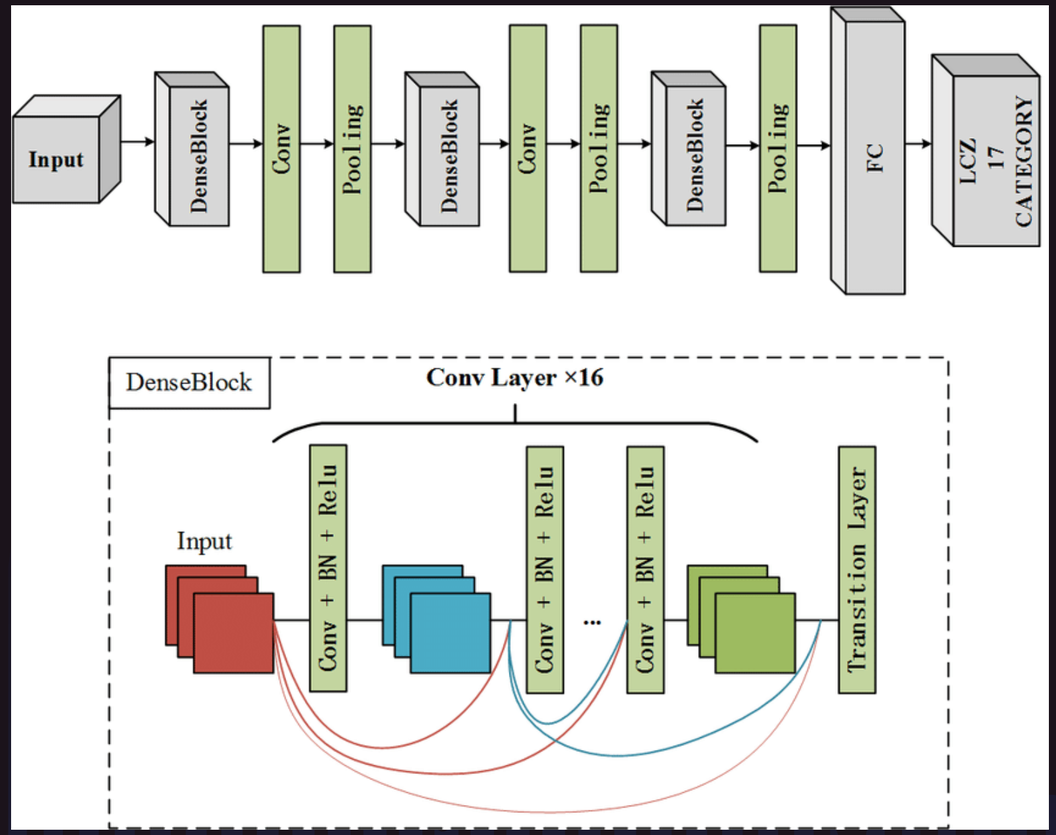
InceptionV3

InceptionV3 is a variant of Inception with 48 layers that uses multiple branches of convolutions with different filter sizes to capture different features. InceptionV3 improves upon Inception by using factorized convolutions, label smoothing, and batch normalization. InceptionV3 can classify images into 1000 categories, such as animals, objects, and scenes. It can also be used as a backbone model for other tasks, such as object detection and segmentation.



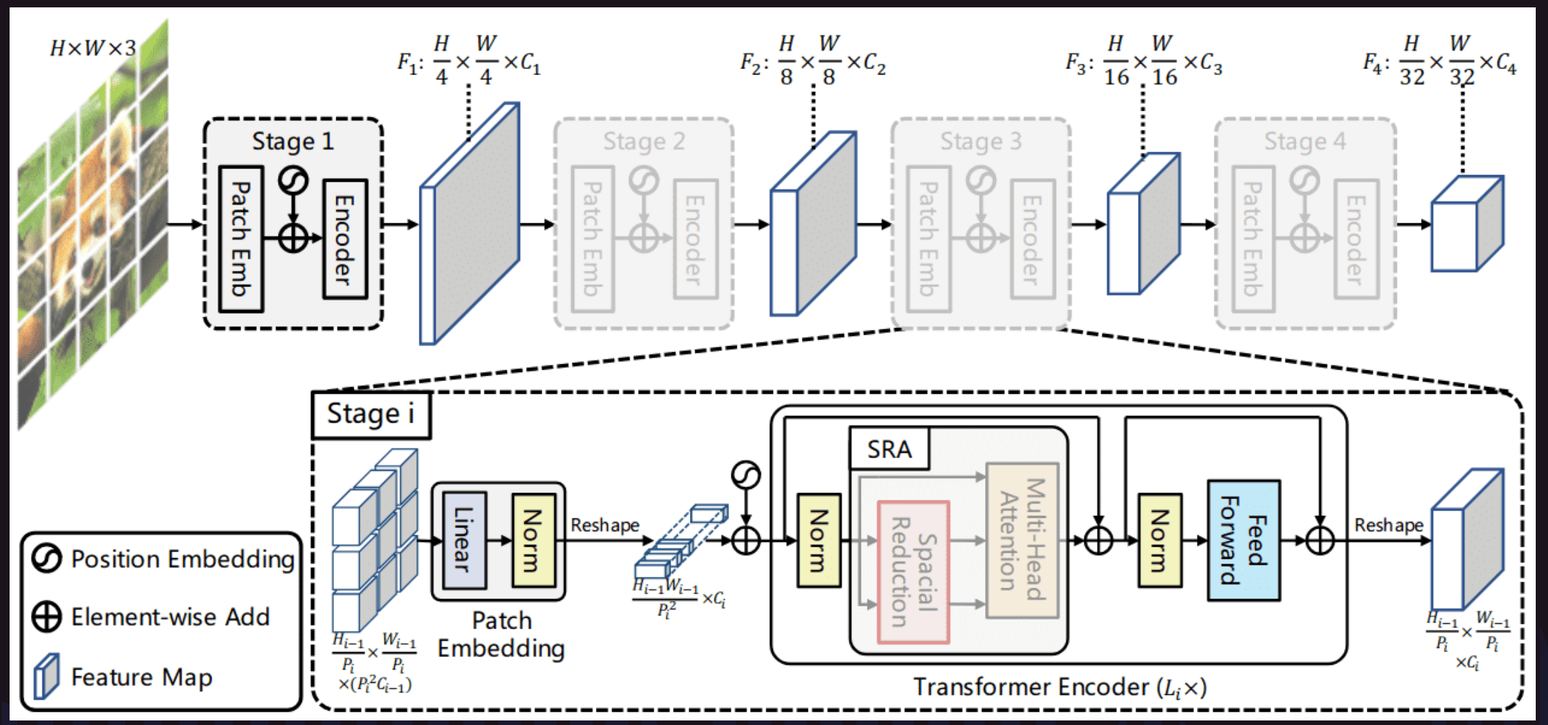
DenseNet201

DenseNet201 is a type DenseNet with 201 layers that uses dense connections between layers. DenseNet201 connects each layer to every other layer in a feed-forward fashion, which helps to alleviate the vanishing-gradient problem, strengthen feature propagation, and reduce the number of parameters. DenseNet201 can classify images into 1000 categories, such as animals, objects, and scenes. It can also be used as a backbone model for other tasks, such as object detection and segmentation.



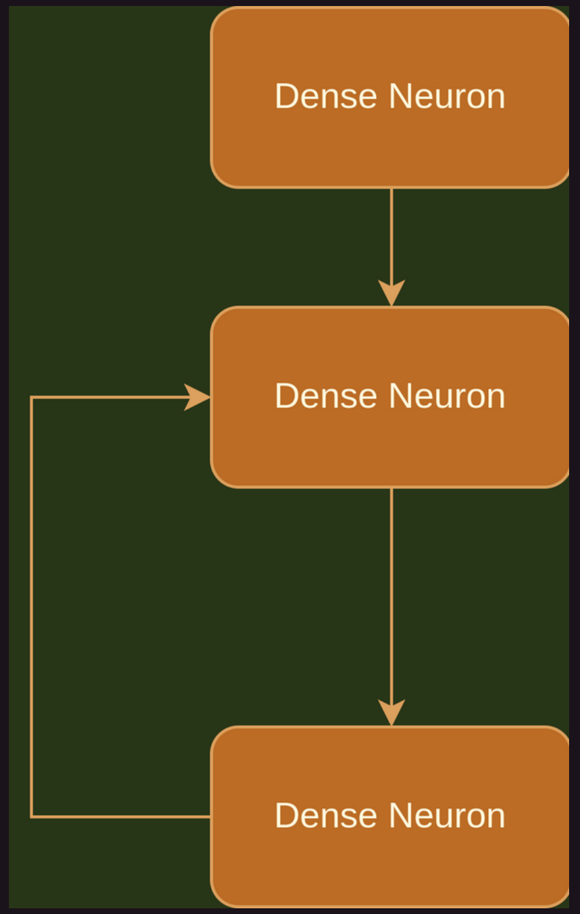
Vision Transformer

The Transformer architecture's applicability to computer vision are still restricted, despite the fact that it has emerged as the de facto standard for natural language processing tasks. Attention is either used in combination with convolutional networks in vision or is used to replace some of the convolutional network's constituent parts while maintaining the overall structure of the network. We demonstrate that a pure transformer applied straight to sequences of picture patches may perform excellently on image classification tasks, eliminating the need for CNN's.



Vision Transformer

Vision Transformer block contains a MLP function.
The function iterates through the hidden_units list and applies a dense layer with the number of hidden units specified by the current element, followed by a GeLU activation function. It then applies a dropout layer with the dropout rate specified by the dropout_rate argument. The purpose of the dropout layer is to randomly drop out some of the neuron activations during training, which can help prevent overfitting and improve the generalization of the model.
The output of the final dense layer is returned as the output of the function.
The motivation behind GELU is to bridge stochastic regularizers, such as dropout, with non-linearities, i.e., activation functions.

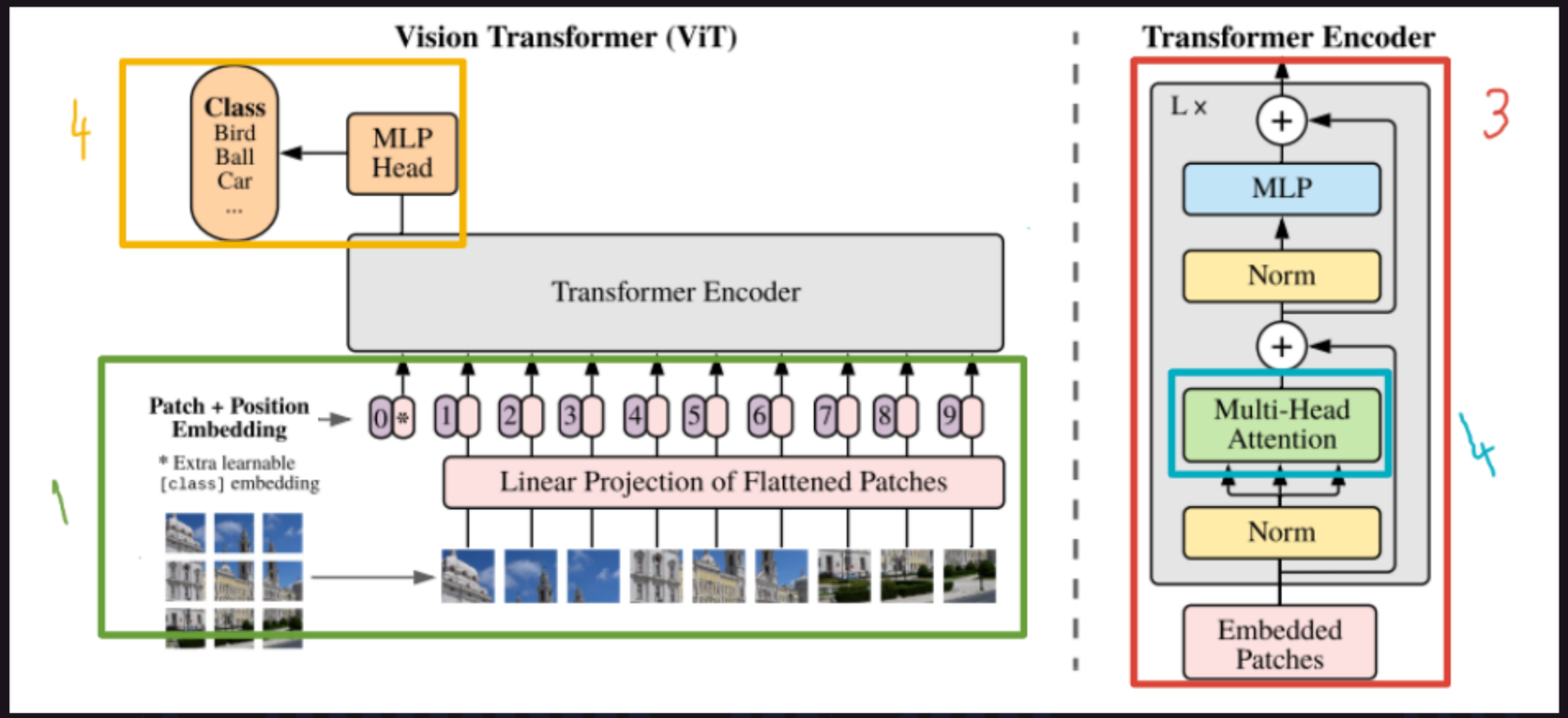


Vision Transformer

The patch encoder that extracts non-overlapping patches of a specified size from an input image. The constructor of the layer takes an integer specifying the size of each patch. The method takes in a batch of images as input, and determines the batch size , then extract patches from each image in the batch. The purpose of this custom layer is to extract patches from input images, which is useful for classification. The extracted patches can be further processed by other layers or fed directly into a neural network. The transformer takes the each patch as if it is a one dimensional text embedding. Finally the transformer proposed by Vaswani et al. is used to obtain the final accuracy.

Video Vision Transformer

Video Vision Transformer (ViT) is a type of deep learning model that applies the Transformer architecture to video classification. It treats each video clip as a sequence of image patches and uses self-attention to learn spatio-temporal features. ViT can achieve state-of-the-art results on various video classification benchmarks, such as Kinetics, Epic Kitchens, and Something-Something v212.



Video Vision Transformer

The tokenization procedure in ViTs involves dividing a picture into patches, which are then spatially flattened. This procedure can be repeated for certain frames in a video. We sample frames from the video clip and use straightforward ViT tokenization in the uniform frame sampling tokenization approach proposed by the authors.

When it comes to extracting temporal information from the video, Tubelet Embedding is unique. We begin by extracting volumes from the video, which include regions of the frame and temporal data. After that, the volumes are flattened to create video tokens. Then positional information to the encoded video tokens.

The authors suggest 4 variants of Vision Transformer:

- Spatio-temporal attention
- Factorized encoder
- Factorized self-attention
- Factorized dot-product attention

Video Vision Transformer

We'll use the Spatio-Temporal Attention Model in this example for the sake of simplicity. The next line of code takes a lot of inspiration from Picture categorisation using Vision Transformer.

The dense neurons and the MLP models which are used in ViVit, is dependant on the rotation of the image. So in the image volume each of the image plate is rotated by 18 degree. The final image is exact mirror opposite of the initial plate. Now when 'Tubelet Embedding' is used, the temporal data is nothing but the representation of the rotational variance of the same image. Then on this image volume the ViVit is used.

We have also tried to use ConvLSTM custom made model on this image volume but the accuracy is obtained was very low 40%. The vision transformer and video vision transformer was by far the best model. We have also used the Vit with genetic algorithm. The estimated training time for ViVit algorithm and the genetic ViT algorithm was 30 hours. It is still running. As of 9 AM today the validation accuracy for ViViT and Genetic Algorithms(1st gen) are percent 87% and 82%.

Video Vision Transformer

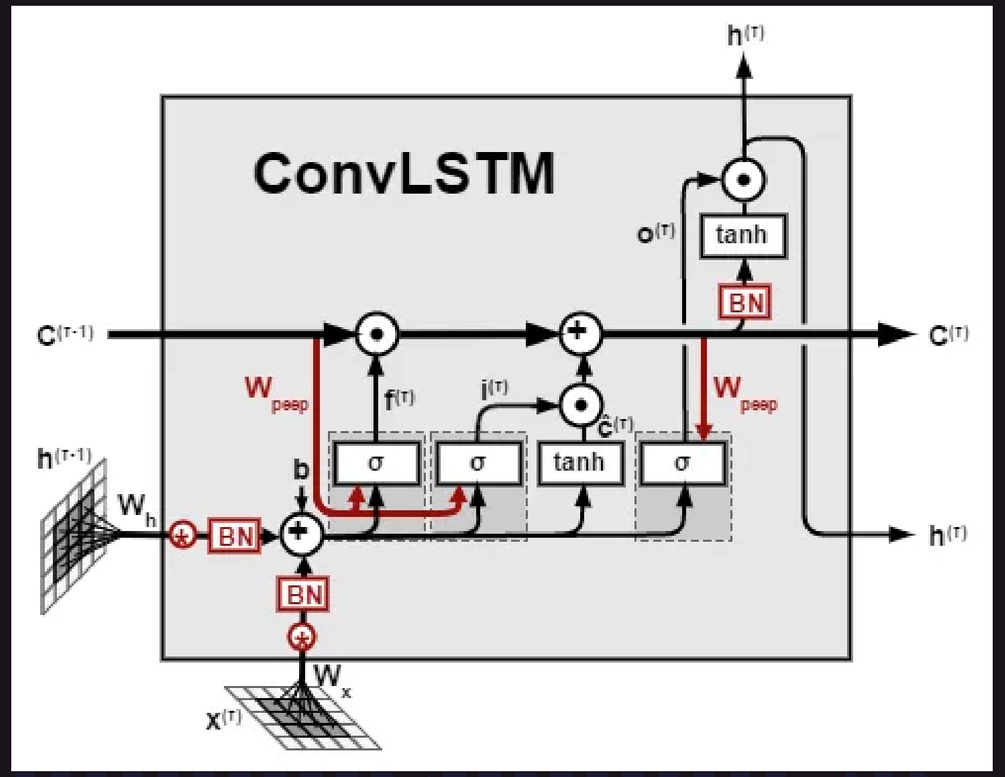
The Transformer architecture's applicability to computer vision are still restricted, despite the fact that it has emerged as the de facto standard for natural language processing tasks. Attention is either used in combination with convolutional networks in vision or is used to replace some of the convolutional network's constituent parts while maintaining the overall structure of the network. We demonstrate that a pure transformer applied straight to sequences of picture patches may perform excellently on image classification tasks, eliminating the need for CNN's.

In our method we have tried to follow the actual transformer, proposed by Vaswani et al., as closely as possible.

We have also imitated the architecture proposed by Dosovitskiy et al. After some fine tuning we have obtained training accuracy 93 %, validation accuracy of 87 % and testing accuracy of 80 %

Convolution LSTM

Convolution LSTM (ConvLSTM) is a type of recurrent neural network that can handle spatio-temporal data, such as videos or weather maps. It combines the advantages of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. ConvLSTM has convolutional structures in both the input-to-state and state-to-state transitions, which allows it to capture spatial dependencies within the input and across time steps. ConvLSTM can be used for tasks such as precipitation nowcasting, video prediction, and action recognition.



General Training Methodology

We used the classification models mention before to test out their applicability for the given task. Many of these models were pretriained on real life data, hence they were not able to deliver accurate result after fine tuning. The worst performing model results at epoch 40 are given below:

VGG16
Training Accuracy: 0.7407
Training precision: 0.7826
Training recall: 0.6779
Training auc: 0.9304
Test Accuracy: 0.5854
Test precision: 0.6207
Test recall: 0.5366
Test auc: 0.8018

ResNet50
Training Accuracy: 0.6079
Training precision: 0.5014
Training recall: 0.2883
Training auc: 0.4865
Test Accuracy: 0.5252
Test precision: 0.3823
Test recall: 0.2504
Test auc: 0.6669

InceptionV3
Training Accuracy: 0.6366
Training precision: 0.7133
Training recall: 0.5166
Training auc: 0.8752
Test Accuracy: 0.5176
Test precision: 0.5659
Test recall: 0.3957
Test auc: 0.7537

DenseNet201
Training Accuracy: 0.7407
Training precision: 0.7826
Training recall: 0.6779
Training auc: 0.9304
Test Accuracy: 0.5854
Test precision: 0.6207
Test recall: 0.5366
Test auc: 0.8018

RESULTS

Content

1. Best Model (ViT) Accuracy
2. Precision
3. Recall
4. F1 Score

Accuracy Report

	Training	Validation
Given Dataset	0.93	0.87
Generated Dataset	0.92	0.73
Combined Dataset	0.86	0.74

Best Model (ViT) Results on Given Dataset

	Angry	Neutral	Happy	Sad	Test Accuracy
Precision	0.82	0.82	0.79	0.82	0.80
Recall	0.88	0.79	0.70	0.88	
F1 Score	0.87	0.80	0.71	0.83	

Best Model (ViT) Results on Combined Dataset

	Angry	Neutral	Happy	Sad	Accuracy
Precision	0.72	0.69	0.83	0.80	0.75
Recall	0.90	0.86	0.51	0.69	
F1 Score	0.80	0.77	0.60	0.77	

Best Model (ViT) Results on Generated Dataset

	Angry	Neutral	Happy	Sad	Accuracy
Precision	0.66	0.77	0.67	0.71	0.72
Recall	0.82	0.81	0.58	0.51	
F1 Score	0.73	0.79	0.62	0.62	

Best Model (ViT) Results on Given Dataset

```
Epoch 39/50
7/7 [==============================] - 37s 5s/step - loss: 0.2618 - accuracy: 0.9104 - top-5-accuracy: 1.0000 - val_loss: 0.4742 - val_accuracy: 0.8477 - val_top-5-accuracy: 1.0000
Epoch 40/50
7/7 [==============================] - 37s 5s/step - loss: 0.3201 - accuracy: 0.8827 - top-5-accuracy: 1.0000 - val_loss: 0.4453 - val_accuracy: 0.8477 - val_top-5-accuracy: 1.0000
Epoch 41/50
7/7 [==============================] - 35s 5s/step - loss: 0.2581 - accuracy: 0.9031 - top-5-accuracy: 1.0000 - val_loss: 0.4444 - val_accuracy: 0.8173 - val_top-5-accuracy: 1.0000
Epoch 42/50
7/7 [==============================] - 35s 5s/step - loss: 0.2487 - accuracy: 0.9070 - top-5-accuracy: 1.0000 - val_loss: 0.4718 - val_accuracy: 0.8274 - val_top-5-accuracy: 1.0000
Epoch 43/50
7/7 [==============================] - 35s 5s/step - loss: 0.2554 - accuracy: 0.9059 - top-5-accuracy: 1.0000 - val_loss: 0.4313 - val_accuracy: 0.8579 - val_top-5-accuracy: 1.0000
Epoch 44/50
7/7 [==============================] - 35s 5s/step - loss: 0.2459 - accuracy: 0.9093 - top-5-accuracy: 1.0000 - val_loss: 0.4664 - val_accuracy: 0.8376 - val_top-5-accuracy: 1.0000
Epoch 45/50
7/7 [==============================] - 35s 5s/step - loss: 0.2168 - accuracy: 0.9195 - top-5-accuracy: 1.0000 - val_loss: 0.4311 - val_accuracy: 0.8528 - val_top-5-accuracy: 1.0000
Epoch 46/50
7/7 [==============================] - 35s 5s/step - loss: 0.2086 - accuracy: 0.9206 - top-5-accuracy: 1.0000 - val_loss: 0.4229 - val_accuracy: 0.8629 - val_top-5-accuracy: 1.0000
Epoch 47/50
7/7 [==============================] - 35s 5s/step - loss: 0.2281 - accuracy: 0.9138 - top-5-accuracy: 1.0000 - val_loss: 0.4829 - val_accuracy: 0.8426 - val_top-5-accuracy: 1.0000
Epoch 48/50
7/7 [==============================] - 35s 5s/step - loss: 0.2234 - accuracy: 0.9223 - top-5-accuracy: 1.0000 - val_loss: 0.4336 - val_accuracy: 0.8426 - val_top-5-accuracy: 1.0000
Epoch 49/50
7/7 [==============================] - 35s 5s/step - loss: 0.2244 - accuracy: 0.9184 - top-5-accuracy: 1.0000 - val_loss: 0.4892 - val_accuracy: 0.8579 - val_top-5-accuracy: 1.0000
Epoch 50/50
7/7 [==============================] - 35s 5s/step - loss: 0.2132 - accuracy: 0.9178 - top-5-accuracy: 1.0000 - val_loss: 0.4414 - val_accuracy: 0.8477 - val_top-5-accuracy: 1.0000
7/7 [==============================] - 1s 181ms/step - loss: 0.6461 - accuracy: 0.7982 - top-5-accuracy: 1.0000
Test accuracy: 79.82%
Test top 5 accuracy: 100.0%
```

Best Model (ViT) Results on Generated Dataset

```
Epoch 38/50
18/18 [==============================] - 92s 5s/step - loss: 0.3423 - accuracy: 0.8728 - top-5-accuracy: 1.0000 -
val_loss: 0.7027 - val_accuracy: 0.7475 - val_top-5-accuracy: 1.0000
Epoch 39/50
18/18 [==============================] - 92s 5s/step - loss: 0.3339 - accuracy: 0.8775 - top-5-accuracy: 1.0000 -
val_loss: 0.7450 - val_accuracy: 0.7414 - val_top-5-accuracy: 1.0000
Epoch 40/50
18/18 [==============================] - 94s 5s/step - loss: 0.3307 - accuracy: 0.8788 - top-5-accuracy: 1.0000 -
val_loss: 0.7016 - val_accuracy: 0.7495 - val_top-5-accuracy: 1.0000
Epoch 41/50
18/18 [==============================] - 93s 5s/step - loss: 0.3211 - accuracy: 0.8847 - top-5-accuracy: 1.0000 -
val_loss: 0.7220 - val_accuracy: 0.7313 - val_top-5-accuracy: 1.0000
Epoch 42/50
18/18 [==============================] - 93s 5s/step - loss: 0.3144 - accuracy: 0.8831 - top-5-accuracy: 1.0000 -
val_loss: 0.7429 - val_accuracy: 0.7495 - val_top-5-accuracy: 1.0000
Epoch 43/50
18/18 [==============================] - 94s 5s/step - loss: 0.2895 - accuracy: 0.8953 - top-5-accuracy: 1.0000 -
val_loss: 0.7251 - val_accuracy: 0.7596 - val_top-5-accuracy: 1.0000
Epoch 44/50
18/18 [==============================] - 93s 5s/step - loss: 0.2838 - accuracy: 0.8944 - top-5-accuracy: 1.0000 -
val_loss: 0.6732 - val_accuracy: 0.7737 - val_top-5-accuracy: 1.0000
Epoch 45/50
18/18 [==============================] - 91s 5s/step - loss: 0.2588 - accuracy: 0.9094 - top-5-accuracy: 1.0000 -
val_loss: 0.6762 - val_accuracy: 0.7475 - val_top-5-accuracy: 1.0000
Epoch 46/50
18/18 [==============================] - 92s 5s/step - loss: 0.2382 - accuracy: 0.9112 - top-5-accuracy: 1.0000 -
val_loss: 0.6903 - val_accuracy: 0.7434 - val_top-5-accuracy: 1.0000
Epoch 47/50
18/18 [==============================] - 93s 5s/step - loss: 0.2326 - accuracy: 0.9189 - top-5-accuracy: 1.0000 -
val_loss: 0.7069 - val_accuracy: 0.7616 - val_top-5-accuracy: 1.0000
Epoch 48/50
18/18 [==============================] - 92s 5s/step - loss: 0.2208 - accuracy: 0.9207 - top-5-accuracy: 1.0000 -
val_loss: 0.7672 - val_accuracy: 0.7475 - val_top-5-accuracy: 1.0000
Epoch 49/50
18/18 [==============================] - 92s 5s/step - loss: 0.2223 - accuracy: 0.9249 - top-5-accuracy: 1.0000 -
val_loss: 0.7326 - val_accuracy: 0.7737 - val_top-5-accuracy: 1.0000
Epoch 50/50
18/18 [==============================] - 92s 5s/step - loss: 0.2161 - accuracy: 0.9227 - top-5-accuracy: 1.0000 -
val_loss: 0.7169 - val_accuracy: 0.7535 - val_top-5-accuracy: 1.0000
18/18 [==============================] - 1s 29ms/step - loss: 0.8577 - accuracy: 0.7436 - top-5-accuracy: 1.0000
Test accuracy: 74.36%
Test top 5 accuracy: 100.0%
```

Best Model (ViT) Results on Combined Dataset

```
Epoch 38/50
12/12 [==============================] - 62s 5s/step - loss: 0.3160 - accuracy: 0.8780 - top-5-accuracy: 1.0000 -
val_loss: 0.8067 - val_accuracy: 0.7440 - val_top-5-accuracy: 1.0000
Epoch 39/50
12/12 [==============================] - 61s 5s/step - loss: 0.2894 - accuracy: 0.8981 - top-5-accuracy: 1.0000 -
val_loss: 0.7589 - val_accuracy: 0.7380 - val_top-5-accuracy: 1.0000
Epoch 40/50
12/12 [==============================] - 61s 5s/step - loss: 0.2821 - accuracy: 0.8927 - top-5-accuracy: 1.0000 -
val_loss: 0.8518 - val_accuracy: 0.7380 - val_top-5-accuracy: 1.0000
Epoch 41/50
12/12 [==============================] - 60s 5s/step - loss: 0.3072 - accuracy: 0.8897 - top-5-accuracy: 1.0000 -
val_loss: 0.8677 - val_accuracy: 0.7139 - val_top-5-accuracy: 1.0000
Epoch 42/50
12/12 [==============================] - 60s 5s/step - loss: 0.2596 - accuracy: 0.9095 - top-5-accuracy: 1.0000 -
val_loss: 0.7912 - val_accuracy: 0.7410 - val_top-5-accuracy: 1.0000
Epoch 43/50
12/12 [==============================] - 62s 5s/step - loss: 0.2702 - accuracy: 0.9071 - top-5-accuracy: 1.0000 -
val_loss: 0.8074 - val_accuracy: 0.7500 - val_top-5-accuracy: 1.0000
Epoch 44/50
12/12 [==============================] - 60s 5s/step - loss: 0.2782 - accuracy: 0.9021 - top-5-accuracy: 1.0000 -
val_loss: 0.8436 - val_accuracy: 0.7199 - val_top-5-accuracy: 1.0000
Epoch 45/50
12/12 [==============================] - 60s 5s/step - loss: 0.2576 - accuracy: 0.8984 - top-5-accuracy: 1.0000 -
val_loss: 0.9060 - val_accuracy: 0.7259 - val_top-5-accuracy: 1.0000
Epoch 46/50
12/12 [==============================] - 60s 5s/step - loss: 0.2636 - accuracy: 0.9045 - top-5-accuracy: 1.0000 -
val_loss: 0.8208 - val_accuracy: 0.7470 - val_top-5-accuracy: 1.0000
Epoch 47/50
12/12 [==============================] - 60s 5s/step - loss: 0.2637 - accuracy: 0.9058 - top-5-accuracy: 1.0000 -
val_loss: 0.8435 - val_accuracy: 0.7380 - val_top-5-accuracy: 1.0000
Epoch 48/50
12/12 [==============================] - 60s 5s/step - loss: 0.2213 - accuracy: 0.9212 - top-5-accuracy: 1.0000 -
val_loss: 0.8672 - val_accuracy: 0.7440 - val_top-5-accuracy: 1.0000
Epoch 49/50
12/12 [==============================] - 60s 5s/step - loss: 0.2127 - accuracy: 0.9246 - top-5-accuracy: 1.0000 -
val_loss: 0.8995 - val_accuracy: 0.7259 - val_top-5-accuracy: 1.0000
Epoch 50/50
12/12 [==============================] - 60s 5s/step - loss: 0.1970 - accuracy: 0.9266 - top-5-accuracy: 1.0000 -
val_loss: 0.7772 - val_accuracy: 0.7289 - val_top-5-accuracy: 1.0000
12/12 [==============================] - 0s 30ms/step - loss: 0.8444 - accuracy: 0.7182 - top-5-accuracy: 1.0000
Test accuracy: 71.82%
Test top 5 accuracy: 100.0%
```

INFERENCE

We observed that the result with given dataset was much better than generated and combined dataset. It was observed that the original dataset was data augmented and using stable diffusion model on the original dataset resulted in more baised dataset. The produced result was degraded for the combined and generated dataset. This may be further improved by the use of ViViT and Genetic ViT Model. The estimated time for training the improved model is 30 hours which is not feasible for this ML Hackathon. The model is still being trained on server. The obtained best accuracy as of 6 AM 24th March of the improved model is 83.11%

THANK YOU