W281 Final Project

Pradeep Chelpati, Jiayi Hu, Nitin Swarup Sokhey

Overview:

The goal of this project is to identify features of images from 20 different categories and develop models to classify them. There are various object or image classification algorithms in deep learning and object detection. Every day, many new models and concepts are added to the current research. The research in deep learning and computer vision is rapidly growing, and every month, the models are being improved. This report is based on image classification using different feature engineering, feature extraction methods using computer vision, dimensionality reduction of the extracted features and implementation of different algorithms on the extracted features to obtain performance reports for different algorithms and to compare these algorithm performances.

Introduction:

The goal of this project is to identify features of images from 20 different categories and develop models to classify them. To classify pictures successfully, we need to make data cleaning, feature exploration, model parameter selection, and model selection. We plan to do the following steps to develop the model and achieve our goal.

1. Extract image features: Different Feature Extraction methods were used in the report to extract different features from the data. The methods used were

- *a) Harris Corner*
- *b) Histogram of Gradients*
- *c) Brisk Feature Extraction*
- *d) CNN-based feature extraction using vgg16*

2. Visualize those features: Most of the features extracted from the images are very high in dimension, and dimensionality reduction algorithms were used. The methods used were

- *a) t-SNE*
- *b) Principal Component Analysis*

3. Train the models: The extracted features were used in training different models. The models which were used in the training of the image data were

- *a) Logistic Regression*
- *b) K-Nearest Neighbors*
- *c) Support Vector Machine*

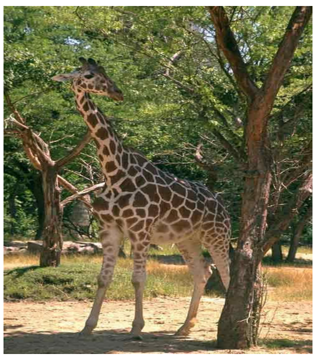
4. Pretrained Model: For advanced extraction of features, a pre-trained model known as VGG16 was used. The features were extracted using the VGG16 layers, which were sent into the custom architecture of a densely connected neural network to get the classification.

Data

We have around 1500 images for training and testing. Sample images are all formatted in JPEG and can be categorized into following categories:
Format: category(number of sample images in this category)
airplane(80), bear(68), blimp(57), comet(81), crab(57), dog(68), dolphin(71), giraffe(56), goat(75), gorilla(141), kangaroo(55), killer-whale(61), leopards(127), llama(79), penguin(99), porcupine(67), teddy-bear(67), triceratops(63), unicorn(65), zebra(64).

Link: https://drive.google.com/drive/folders/1iZXE-zPETqsN9TpnUIOy07gzJPPYIwRh

Example images:





Methodology

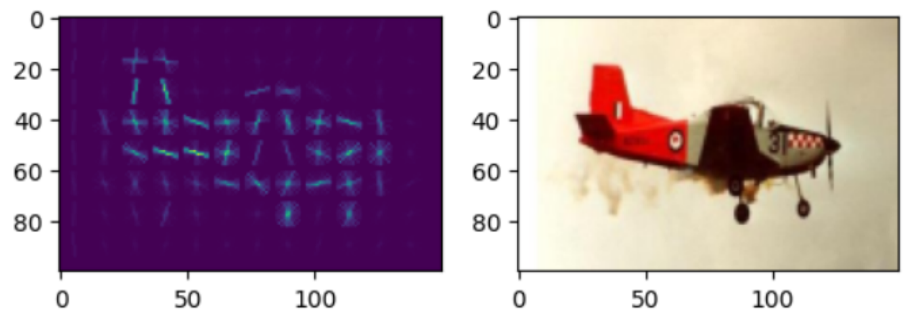
Feature Extraction:

We used 4 methods for feature extraction.

*a) **Histogram of Oriented Gradients:***

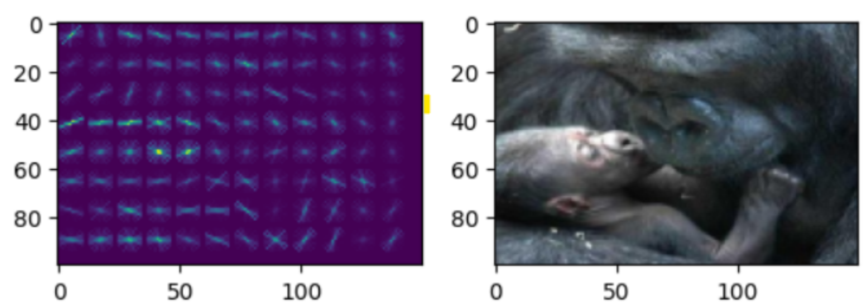
A histogram of Oriented gradients counts the number of occurrences of gradient orientation in local parts of an image. As it gives features which are invariant to any kind of geometric changes or photometric changes, this feature extraction method was used in extracting features from the image (Song et al., 2014).

To enhance the contrast of the image using LAB color space, the equalization using the histogram on the channel L and, after adaptive histogram equalization, converting the image back to the BGR results in enhanced brightness with limited sensitivity in contrast.



Inference:

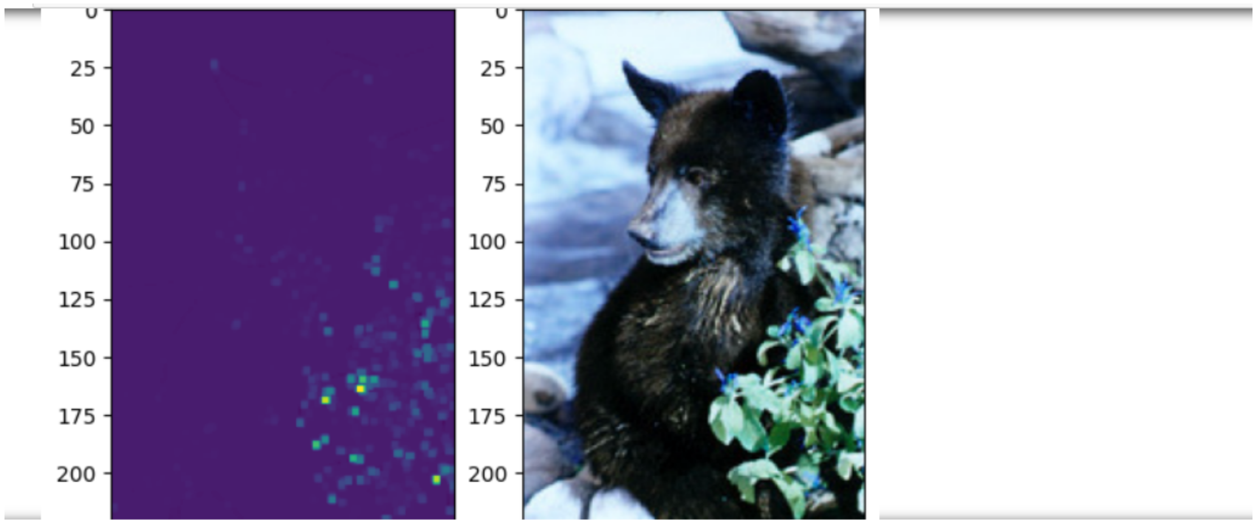
For the above image, the histogram of gradients can be recognized easily as the background and foreground are different, and there are not many objects in the image.



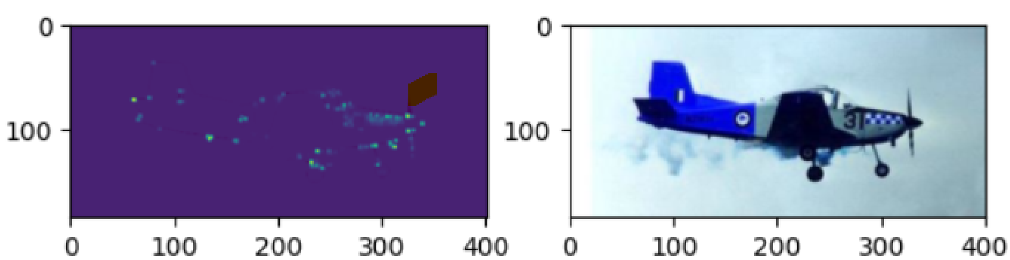
For the above image, the image has several objects hence contours need to be recognized properly.

b) Harris Corner Features:

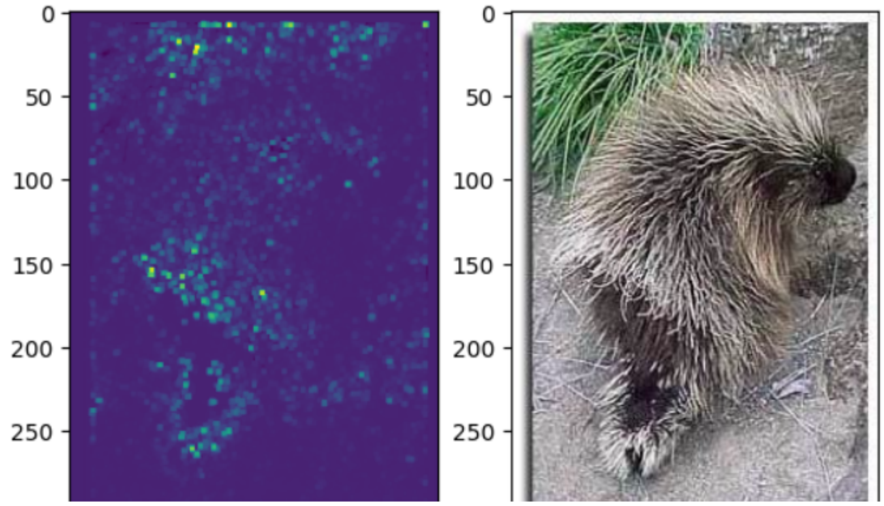
As our dataset mostly has objects and animals, the Harris corner detection will be a very good method to extract meaningful features which can be used in the image classification (Gao et al., 2013).



For the above image, the corners need to be recognized properly.



For the airplane image above, the corners are appropriately recognized. These corners can be good features for image classification as the corners will sometimes be different for a different object, which can become a distinguishable feature.

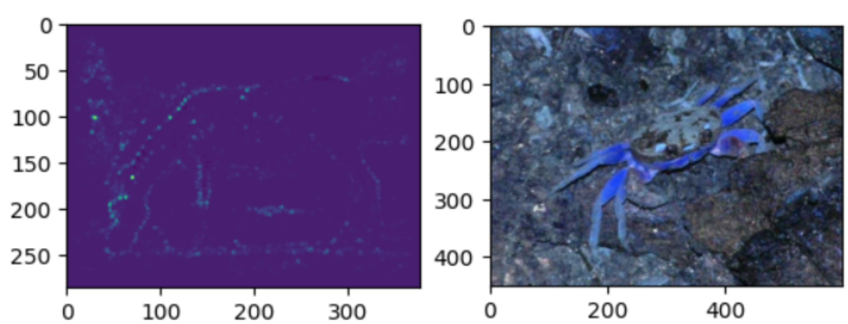


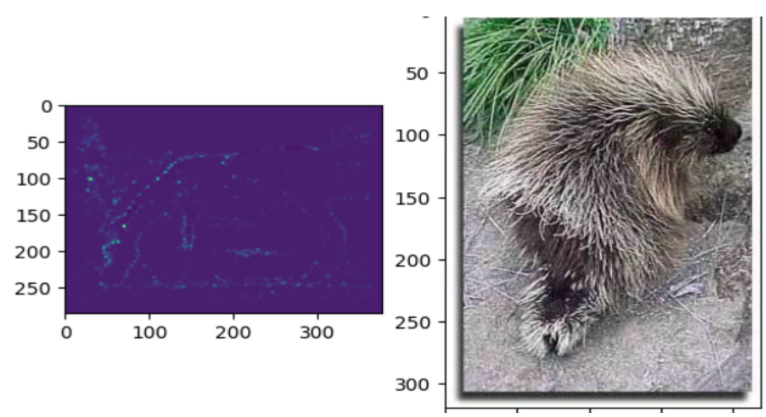
In the above image, the grass and the porcupine are having the same grass-like features, the harris corner needs to be fixed on the above image.

c) Brisk Feature Extraction:

The BRISK feature extraction method was used as it rapidly extracts the key points in the images and the features extracted are invariant to rotation, scale and noise (Li & Chen, 2018).

Examples of BRISK Features:





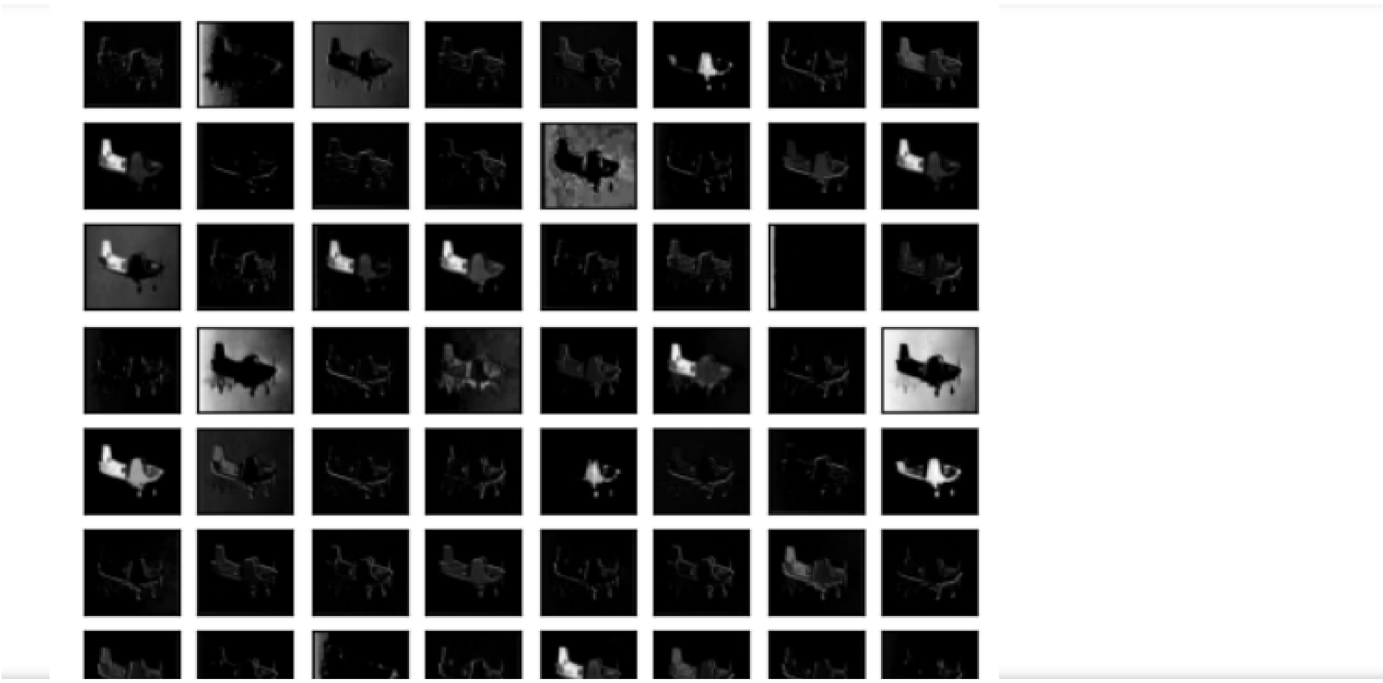
As seen in the above images, the BRISK feature extraction did well on images like a porcupine and blue crab. Harris corner was not able to recognize corners in these images as well as the BRISK.

d) Convolution Based Feature Extraction:

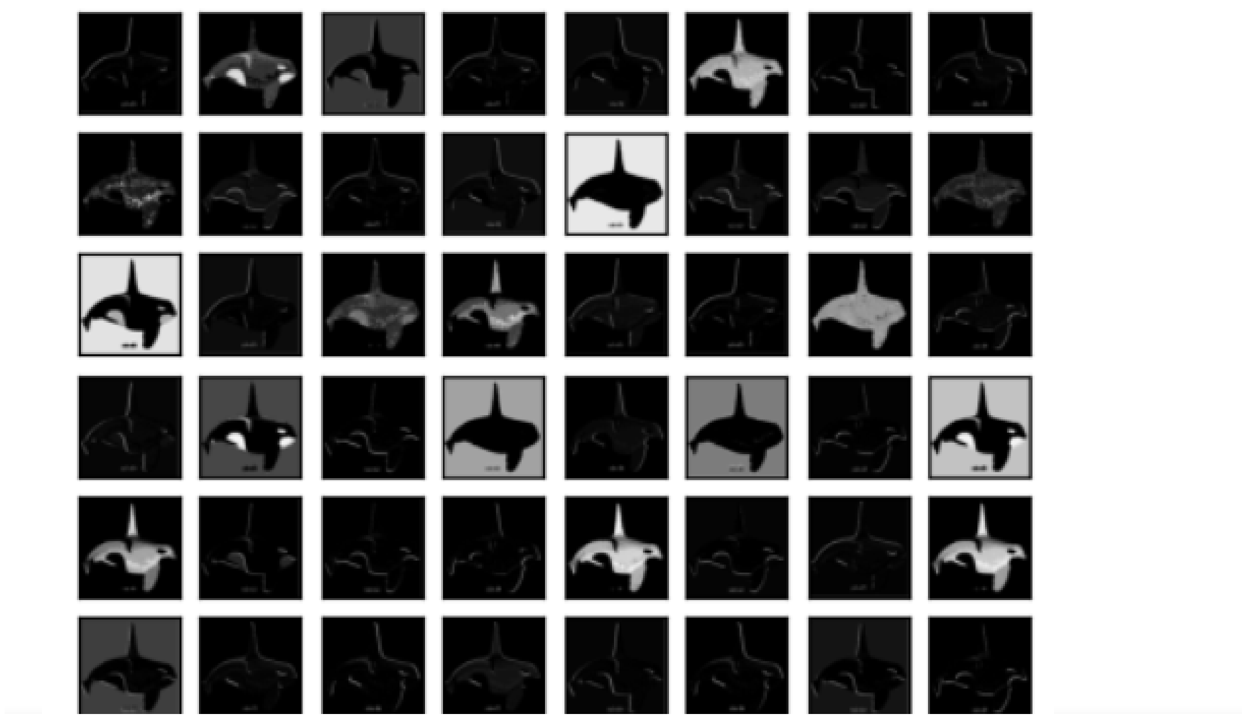
Convolution is an operation where the kernel is flipped, and the features are extracted. These features extracted depend on the filter or kernel used while convolving on the image. The convolution operation can get spatial invariant features which are usually not detected or extracted by usual computer vision methods (Figure 4: (A) Architecture of the Original VGG16, (B) VGG16 Architecture with the Strategy Applied., n.d.).

In this report, the convolution features were extracted using a pre-trained model such as VGG16.

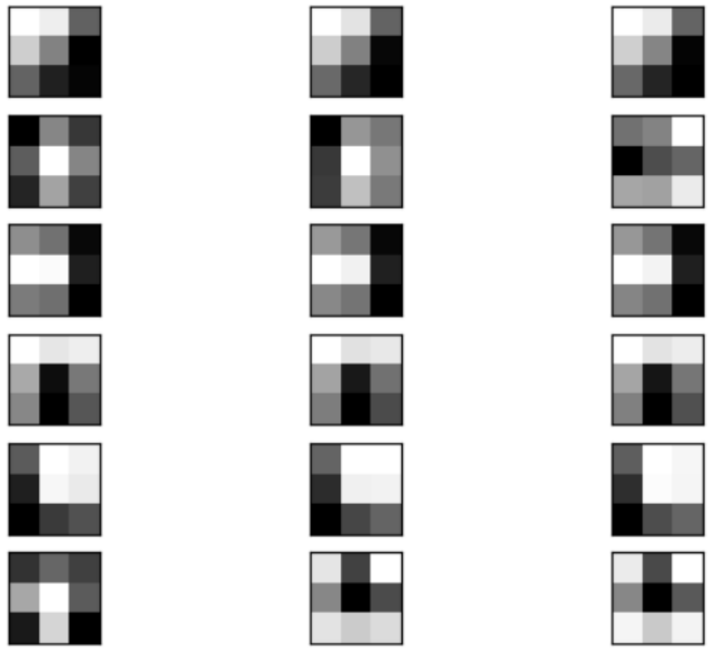
Examples of features Extracted:



Features Extracted for airplane-based images



Features extracted for dolphin-based images



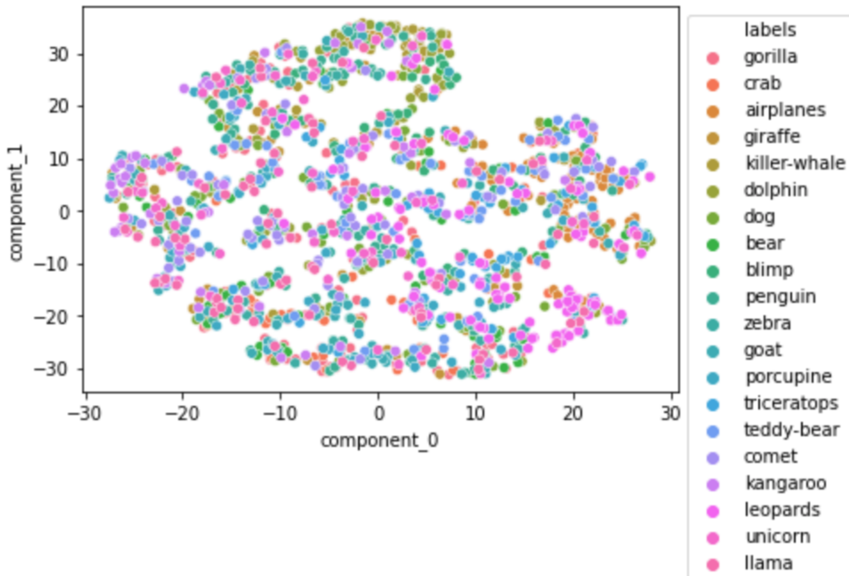
Types of features extracted.

Dimensionality Reduction

a) ***Principal Component Analysis:***

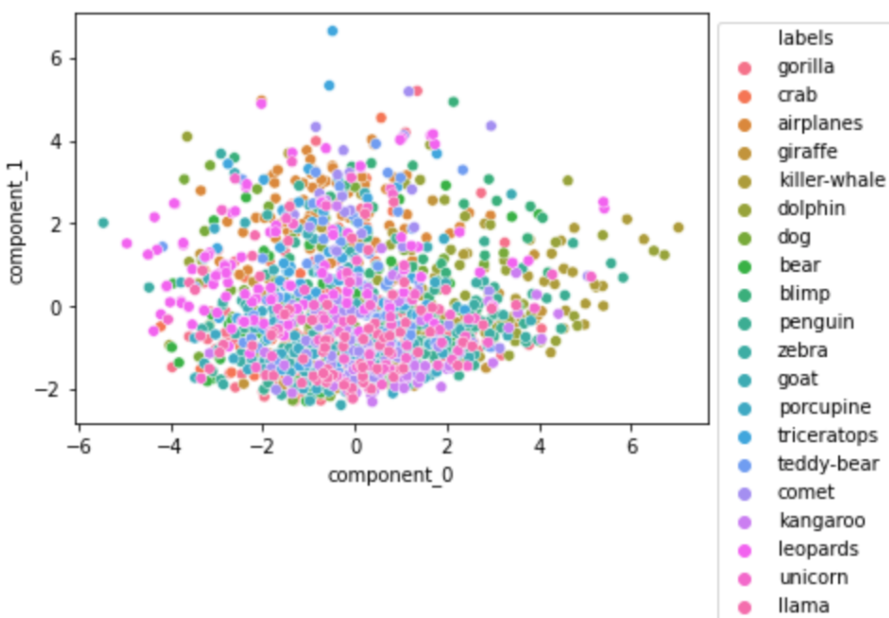
Principal Component Analysis is a method which uses Eigen vectors and Eigen values to reduce the linearly dependent features in the data, and by doing this, it reduces the dimensionality of the data. As there were many features during the feature extraction step, many features caused a dimensionality problem in many algorithms, hence principal component analysis was used to reduce dimensions (Sanguansat, 2012).

PCA Visualization:



b) t-SNE

The t-SNE was used on top of the Principal Component Analysis to reduce the data dimensionality further. t-SNE is good at dimensionality reduction but it is too expensive to be used computationally, hence first the features were reduced using PCA and then sent into t-SNE to further reduce the dimensionality (T-SNE, n.d.).



Modelling:

For Modelling, the following models were used.

The features used for modelling phase 1 were ***Histogram of oriented gradients:***

1) ***Logistic Regression:***

Logistic regression is a model which is based on the sigmoid function. It is based on the log of odds function. When using logistic regression, the model gives the probability of an event happening. Mostly logistic regression is preferred for binary classification but also can be used for multi-class classification where there could be some modification done to adopt logistic regression to multi-class by using one vs. one or one vs. rest-based strategies ("Multiple Logistic Regression," 2005).

2) ***K Nearest Neighbors:***

This algorithm is non-parametric. It is mainly used in supervised learning. It is also used in ensemble-based algorithms. In KNN, the new data point is given a prediction based on the neighbours near the data point. For regression, there is mostly averaging of the values for the neighbours. For classification, the majority voting of the nearest neighbours determines the prediction of the new data point. There are many distance metrics involved when using KNN, the most common metric is euclidean distance. The other metrics used in the KNN are manhattan distance, Minkowski distance, and grovers distance. Mainly KNN is applied in Finance, Healthcare, and in pattern recognition (Classification: Theory - KNN, 2018).

3) ***Support Vector Machine:***

This algorithm is parametric. It is mainly used in supervised learning. It is also used in ensemble-based algorithms. In SVM, the new data point is given a prediction based on the decision boundary near the data point. The SVM chooses the decision boundary based on the data points which make up the hyperplane. The SVM can identify both linear and non-linear relationships. The SVM can identify non-linear relationships by using kernel tricks. The kernel trick involves increasing or transforming lower dimensional space into higher dimensional space to identify relationships. The kernels used in the SVM are linear, polynomial, radial basis, and exponential kernels (Classification: Theory - KNN, 2018).

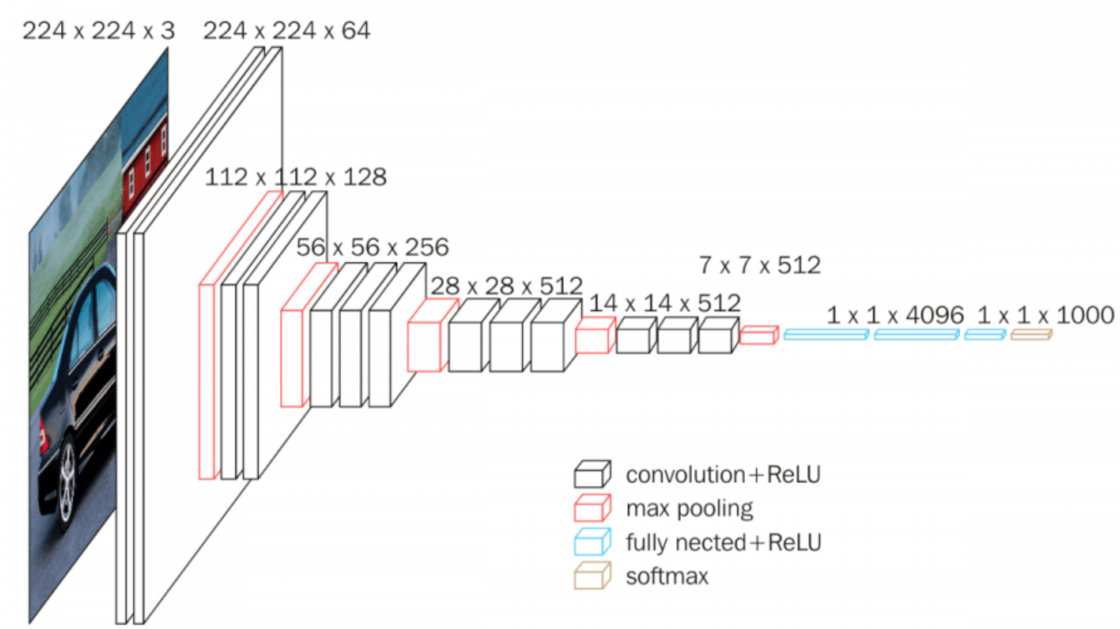
For Modelling Phase 2,

The advanced strategies used in modelling Phase 2 includes the pre-trained models.

a) VGG16:

The VGG16 is a convolution-based neural network which was trained on millions of images from a large database known as ImageNet. This pre-trained network can classify images into 1000 image categories (Figure 4: (A) Architecture of the Original VGG16, (B) VGG16 Architecture with the Strategy Applied., n.d.).

VGG16 Architecture:



Source:
https://neurohive.io/wp-content/uploads/2018/11/vgg16-1-e1542731207177.png

Then VGG 16 was attached to the custom-made densely connected neural network.

Results:

a) Logistic Regression:

Train:

	precision	recall	f1-score	support
airplanes	1.00	1.00	1.00	64
bear	1.00	1.00	1.00	54
blimp	1.00	1.00	1.00	45
comet	1.00	1.00	1.00	55
crab	1.00	1.00	1.00	45
dog	1.00	1.00	1.00	54
dolphin	1.00	1.00	1.00	56
giraffe	1.00	1.00	1.00	45
goat	1.00	1.00	1.00	60
gorilla	1.00	1.00	1.00	112
kangaroo	1.00	1.00	1.00	44
killer-whale	1.00	1.00	1.00	49
leopards	1.00	1.00	1.00	102
llama	1.00	1.00	1.00	63
penguin	1.00	1.00	1.00	79
porcupine	1.00	1.00	1.00	54
teddy-bear	1.00	1.00	1.00	54
triceratops	1.00	1.00	1.00	50
unicorn	1.00	1.00	1.00	52
zebra	1.00	1.00	1.00	51
accuracy			1.00	1188
macro avg	1.00	1.00	1.00	1188
weighted avg	1.00	1.00	1.00	1188

Inference:

The model was able to learn with an accuracy of 100 on all the training images.

Test:

	precision	recall	f1-score	support
airplanes	1.00	0.94	0.97	16
bear	0.00	0.00	0.00	14
blimp	0.50	0.36	0.42	11
comet	0.40	0.43	0.41	14
crab	0.22	0.18	0.20	11
dog	0.18	0.14	0.16	14
dolphin	0.20	0.14	0.17	14
giraffe	0.00	0.00	0.00	11
goat	0.00	0.00	0.00	15
gorilla	0.25	0.50	0.33	28
kangaroo	0.50	0.36	0.42	11
killer-whale	0.14	0.25	0.18	12
leopards	0.65	0.96	0.77	25
llama	0.31	0.25	0.28	16
penguin	0.21	0.30	0.25	20
porcupine	0.18	0.15	0.17	13
teddy-bear	0.38	0.38	0.38	13
triceratops	0.45	0.38	0.42	13
unicorn	0.25	0.08	0.12	13
zebra	0.43	0.46	0.44	13
accuracy			0.35	297
macro avg	0.31	0.31	0.30	297
weighted avg	0.33	0.35	0.33	297

Inference:

Although, the model performed with 100 percent accuracy on the training data, on the test data it was only able to perform well without any overfitting on the airplanes and performed average on blimps, comets, zebra and leopards but didn't perform well on other objects, Hence, a lot of overfitting was seen in this model.

b) Support Vector Machine

Train:

	precision	recall	f1-score	support
airplanes	1.00	1.00	1.00	64
bear	1.00	0.98	0.99	54
blimp	1.00	1.00	1.00	45
comet	1.00	1.00	1.00	55
crab	1.00	0.96	0.98	45
dog	1.00	1.00	1.00	54
dolphin	1.00	0.98	0.99	56
giraffe	1.00	1.00	1.00	45
goat	1.00	1.00	1.00	60
gorilla	0.92	1.00	0.96	112
kangaroo	1.00	0.93	0.96	44
killer-whale	1.00	0.98	0.99	49
leopards	0.95	1.00	0.98	102
llama	1.00	1.00	1.00	63
penguin	1.00	0.99	0.99	79
porcupine	1.00	0.96	0.98	54
teddy-bear	1.00	1.00	1.00	54
triceratops	1.00	0.96	0.98	50
unicorn	1.00	0.98	0.99	52
zebra	1.00	0.98	0.99	51
accuracy			0.99	1188
macro avg	0.99	0.99	0.99	1188
weighted avg	0.99	0.99	0.99	1188

Inference:

The model was able to perform with 99 percent accuracy on training data.

Test:

	precision	recall	f1-score	support
airplanes	1.00	0.94	0.97	16
bear	0.00	0.00	0.00	14
blimp	0.67	0.18	0.29	11
comet	1.00	0.14	0.25	14
crab	0.00	0.00	0.00	11
dog	0.00	0.00	0.00	14
dolphin	0.00	0.00	0.00	14
giraffe	0.00	0.00	0.00	11
goat	0.00	0.00	0.00	15
gorilla	0.14	0.89	0.24	28
kangaroo	0.00	0.00	0.00	11
killer-whale	0.38	0.25	0.30	12
leopards	0.65	0.96	0.77	25
llama	0.00	0.00	0.00	16
penguin	0.12	0.20	0.15	20
porcupine	0.00	0.00	0.00	13
teddy-bear	0.25	0.15	0.19	13
triceratops	0.60	0.46	0.52	13
unicorn	0.00	0.00	0.00	13
zebra	0.00	0.00	0.00	13
accuracy			0.28	297
macro avg	0.24	0.21	0.18	297
weighted avg	0.25	0.28	0.22	297

Inference:

Although, the model performed with 99 percent accuracy on the training data, on the test data it was only able to perform well without any overfitting on the airplanes and performed average on blimps, comets, zebra and leopards but didn't perform well on other objects, Hence, a lot of overfitting was seen in this model.

c) K-Nearest Neighbors:

Train:

	precision	recall	f1-score	support
airplanes	0.92	0.95	0.94	64
bear	0.30	0.83	0.44	54
blimp	0.88	0.67	0.76	45
comet	0.21	0.85	0.34	55
crab	0.30	0.62	0.41	45
dog	1.00	0.33	0.50	54
dolphin	0.24	0.48	0.32	56
giraffe	0.17	0.47	0.25	45
goat	0.42	0.17	0.24	60
gorilla	0.85	0.21	0.33	112
kangaroo	0.48	0.27	0.35	44
killer-whale	0.65	0.31	0.42	49
leopards	0.52	1.00	0.69	102
llama	0.77	0.16	0.26	63
penguin	0.55	0.15	0.24	79
porcupine	0.82	0.26	0.39	54
teddy-bear	0.73	0.15	0.25	54
triceratops	1.00	0.16	0.28	50
unicorn	1.00	0.04	0.07	52
zebra	1.00	0.04	0.08	51
accuracy			0.42	1188
macro avg	0.64	0.41	0.38	1188
weighted avg	0.65	0.42	0.39	1188

Inference:

The model performed with 42 percent accuracy on the training data, on the training data it was only able to perform well on the airplanes and performed average on blimps, comets, zebra and leopards but didn't perform well on other objects,

Test:

	precision	recall	f1-score	support
airplanes	1.00	0.94	0.97	16
bear	0.10	0.29	0.15	14
blimp	0.00	0.00	0.00	11
comet	0.12	0.50	0.19	14
crab	0.29	0.45	0.36	11
dog	0.00	0.00	0.00	14
dolphin	0.19	0.43	0.26	14
giraffe	0.08	0.27	0.12	11
goat	0.00	0.00	0.00	15
gorilla	0.00	0.00	0.00	28
kangaroo	0.00	0.00	0.00	11
killer-whale	0.15	0.17	0.16	12
leopards	0.43	1.00	0.60	25
llama	0.00	0.00	0.00	16
penguin	0.00	0.00	0.00	20
porcupine	0.40	0.15	0.22	13
teddy-bear	0.50	0.08	0.13	13
triceratops	0.50	0.15	0.24	13
unicorn	0.00	0.00	0.00	13
zebra	0.00	0.00	0.00	13
accuracy			0.24	297
macro avg	0.19	0.22	0.17	297
weighted avg	0.19	0.24	0.18	297

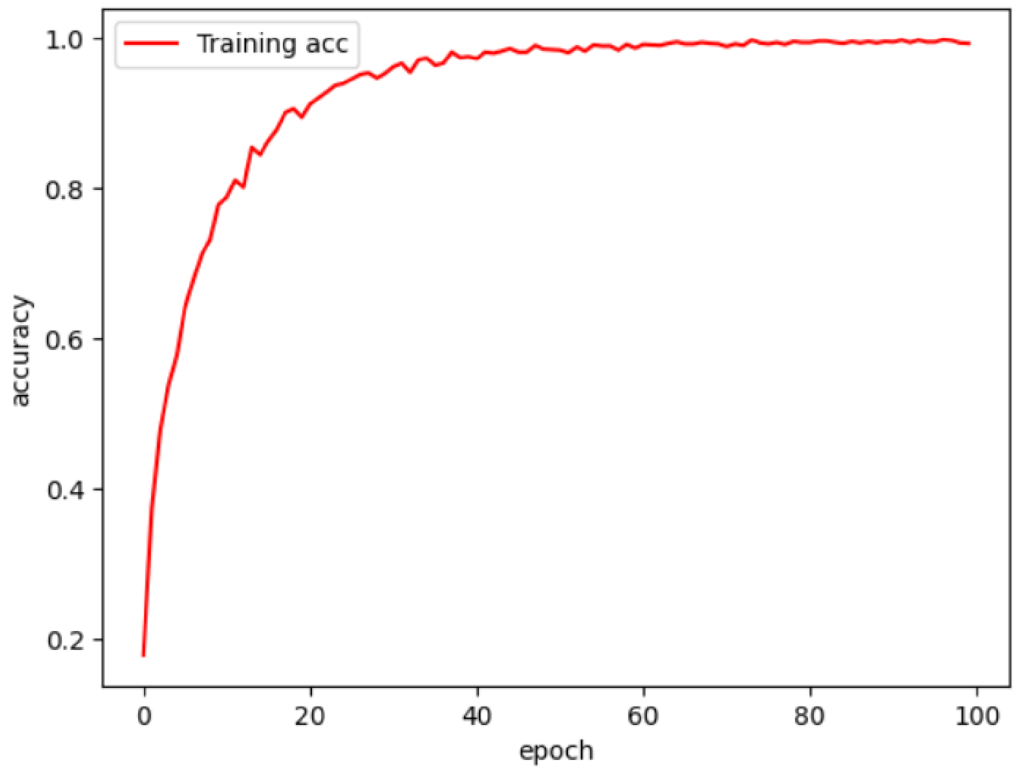
Inference:

Although, the model performed with 42 percent accuracy on the training data, on the test data it was only able to perform well without any overfitting on the airplanes and performed average on blimps, comets, zebra and leopards but didn't perform well on other objects, Hence, some overfitting was seen in this model.

VGG-16:

The VGG-16 was used on training data and it was tested using the testing data provided in the dataset. The training accuracy for VGG16 was 0.9975.

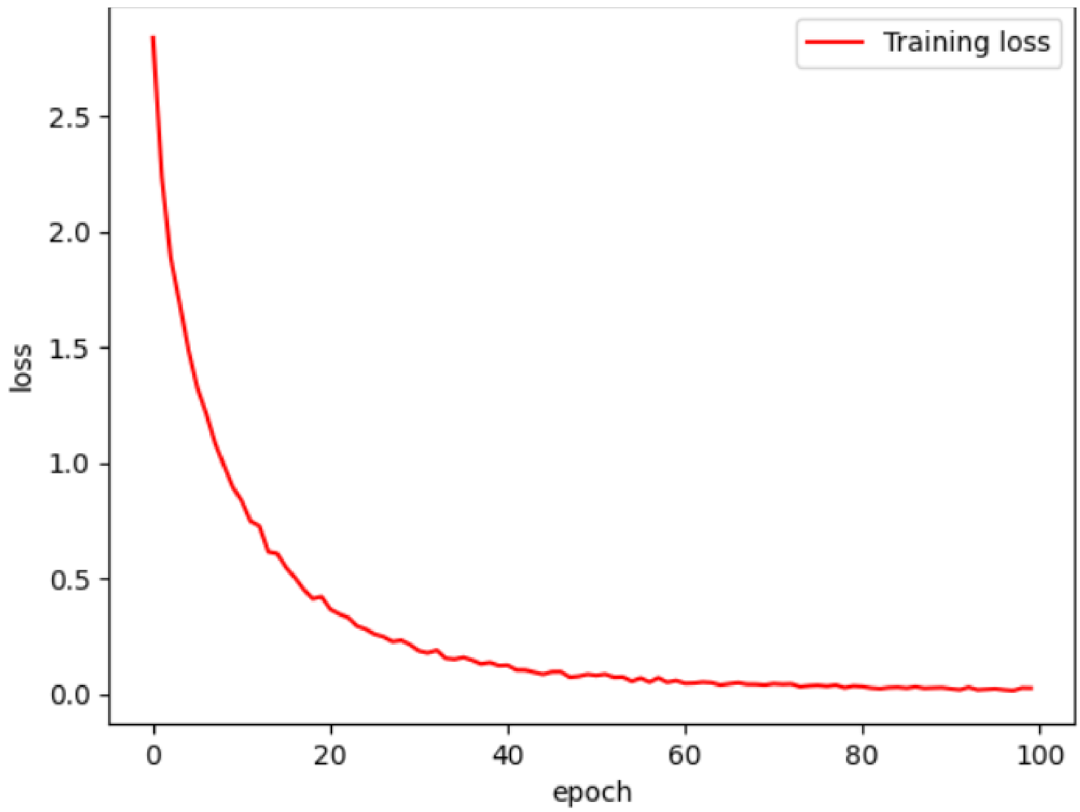
The training loss vs epoch graph:



Inference:

It can be seen from the above graph that there was an exponential increase in the training accuracy from 0 to 16 epochs.

Training Loss Graph



Inference:

From the above image it can be seen that the loss exponentially decreased from 0 to 16 epochs which validates the graph for training accuracy vs epochs.

Conclusion:

In the report, various feature extraction methods were explored and the features extracted were dimensionally reduced into lesser dimensional space using PCA followed by t-SNE.

First, the histogram of gradient features was extracted from the images. Then these features were used in the logistic regression model and support vector machine model to classify all the images. The Logistic regression model needed to perform better on the given features. The Logistic regression was able to perform with a training performance of 100 percent accuracy, but the test performance was 35 percent. Which showed clear overfitting. The SVM performed not *slightly better than Logistic Regression, but still, there was prominent overfitting. The KNN with neighbors of 3 didn’t perform better than Logistic regression and SVM.*

Then the pre-trained model vgg16 was used for training the data. The VGG16 performed better than the rest of the models as the vgg16 was trained on a larger amount of data previously and those weights were used in training the current data. VGG had a training accuracy of approximately 99 percent. As VGG used convolution-based filters to extract complex features, it performed better than any other algorithms. The test data was predicted using the vgg16 and results were stored in results.

References

Classification: Theory - KNN. (2018) - https://doi.org/10.4135/9781526469120

Figure 1: SVM classifier. (A) SVM classification technique. (B) SVM hyperplane selection. (n.d.). https://doi.org/10.7717/peerj-cs.437/fig-1

Figure 4: (A) Architecture of the original VGG16, (B) VGG16 architecture with the strategy applied. (n.d.). https://doi.org/10.7717/peerj-cs.451/fig-4

Gao, Q., Xu, P., & Yang, L. (2013). Breakage detection for grid images based on improved Harris corner. Journal of Computer Applications, 32(3), 766–769. - https://doi.org/10.3724/sp.j.1087.2012.00766

Li, Z., & Chen, C. (2018, November). An improved adaptive threshold BRISK feature matching algorithm based on SURF. 2018 Chinese Automation Congress (CAC). - http://dx.doi.org/10.1109/cac.2018.8623194

Multiple logistic regression. (2005). In Applied Logistic Regression (pp. 31–46). John Wiley & Sons, Inc. - http://dx.doi.org/10.1002/0471722146.ch2

Sanguansat, P. (2012). Two-Dimensional principal component analysis and its extensions. In Principal Component Analysis. InTech. - http://dx.doi.org/10.5772/36892

Song, D., Tang, L., & Zhao, B. (2014). The object recognition algorithm based on affine histogram of oriented gradient. Journal of Electronics & Information Technology, 35(6), 1428–1434. - https://doi.org/10.3724/sp.j.1146.2012.01241

t-SNE. (n.d.). Laurens van Der Maaten. Retrieved December 12, 2022 - https://lvdmaaten.github.io/tsne/