1: Table of Contents

2: Executive Summary

The goal of our project was to compare car recognition accuracy between convolutional networks and systems that used traditional computer vision methods. We also aimed to compare how their accuracy changed when they were tested with different augmented images.

To achieve this goal, we firstly prepared a dataset of 16663 grayscale images, part of which contained cars and the rest of which did not. This dataset would be used as input for three implementations. One of these was AlexNet, a convolutional neural network. The other two were visual bags of words and histogram of features, which are based on traditional computer vision methods. We split the dataset to reserve a set of images for testing, then used the rest of the images to train our image recognition systems. After doing that, we used the validation set to determine the best model and to collect accuracy metrics for each system using the test set of images. We also used a set of augmented images to observe how those metrics changed when the systems were tested with augmented images.

Visual bags of words was a system that firstly collected feature descriptors from each image and clustered them. We chose FREAK and ORB descriptors for this system because of their invariance to certain transformations. After clustering, the images' extracted feature descriptors were converted into frequency histograms of assigned cluster labels. The histograms were then used as inputs to a machine learning classifier to recognize whether an input image contained cars or not..

Histogram of features was a system that firstly discretized each image into 64 grids and extracted a feature descriptor from each grid with its centroid as the keypoint. After representing each feature descriptor as an intensity histogram, we concatenated the histograms to form a single histogram representing the entire image. The histograms were used as input to a machine learning classifier for it to recognize images as containing cars or not containing cars. We experimented with using PCA to decrease dimensionality of the features and increase accuracy of the machine learning classifier.

When comparing the accuracy metrics of the systems, we found that AlexNet tended to have higher accuracy metrics than the traditional computer vision recognition systems. However, a histogram of features-based system using HOG features and PCA transformed features resulted in accuracy metrics that were close to AlexNet's. AlexNet's ROC AUC score was 0.991, while the HOG histogram of features' ROC AUC score was 0.959. From this, we concluded that AlexNet was more accurate than systems using traditional computer vision methods, but a HOG histogram of features system using PCA transformed features was comparable to AlexNet.

When exposed to augmented images, all three systems' accuracy metrics decreased by different amounts. In the case of mixed augmentations, AlexNet's accuracy metrics were still higher than those of visual bags of words and histogram of features, whose metrics had decreased less sharply. From this comparison, we firstly concluded that AlexNet was more robust to augmentation than the other systems. Because all systems suffered a decrease in accuracy after being exposed to augmented images, we also concluded that the classification

accuracy of these recognition systems depends largely on the features it has learned. The systems were less accurate when tested with augmented images because the images contained features they had not learned.

3: Introduction

The field of object recognition has recently changed. Less than a decade ago, most image recognition methods depended on a combination of traditional computer vision and machine learning methods, until the arrival of AlexNet, which used a convolutional neural network. In 2012, AlexNet was submitted for the ImageNet Large Scale Visual Recognition Challenge and won. At the time, it was the most successful object recognition system to use convolutional neural networks, with an error rate of more than 10 percent lower than its runner-up (ImageNet, 2012). Convolutional networks have been popular in image recognition implementations ever since.

Convolutional networks' increased use and lower error rates leave one to wonder what merit the traditional computer vision methods have for object recognition nowadays. More specifically, it would be beneficial to know whether convolutional networks have flaws that traditional computer vision methods do not. In addition, because convolutional networks rely on their datasets for classification quality, it would also be good to know how robust they are to images they were not exposed to. These were the primary motivations for our project. The goal of our project is to compare car recognition accuracy of convolutional networks to that of systems that use traditional computer vision methods such as feature detection. We also want to compare how their accuracy changes when tested with augmented images.

4: Approach

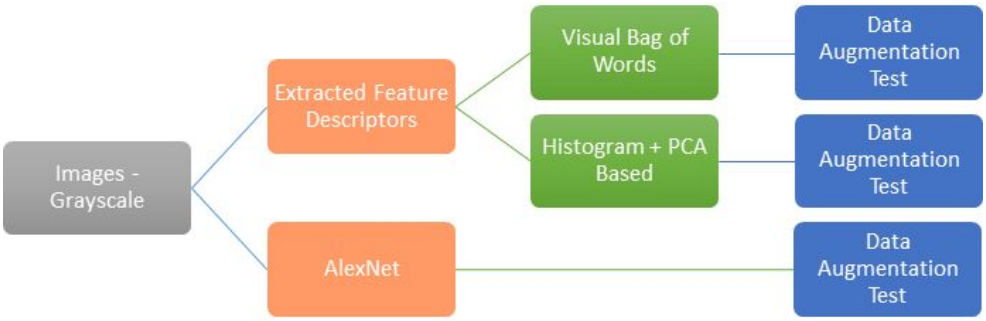

Figure 1: A diagram illustrating our overall methodology.

To achieve the goal of our project, we firstly prepared a dataset of images. For classification purposes, this dataset consisted of two types of images: images that contained cars in different orientations, and images that did not contain any cars. We preprocessed these images by converting them to grayscale before using them.

We used the dataset images as training input for an AlexNet implementation. We then extracted feature descriptors from the images using three types of feature detectors: ORB, FREAK, and HOG. Before using HOG, we applied a Gaussian filter to the image to reduce noise.

DESCRIPTORS' INVARIANCE

	Scale	Rotation	Viewpoint	Lightness
SIFT	x	x	x	x
SURF	x	x	x	x
DAISY			x	x
BRIEF				x
ORB		x		x
BRISK	x	x		x
FREAK	x	x	x	x
LATCH		x		x

Figure 2: Table showing invariance of the descriptors to different transformations (Chatoux, 2016).

We chose these types of feature descriptors because of their resilience to different transformations, quick speeds, and due to their ability to retain crucial information for efficient image description. ORB is invariant to rotation and is resistant to noise. FREAK is resistant to scaling, rotation, and translation due to its unique way of sampling near the keypoint. Although HOG is not transformation invariant, it captures edge information very well, which can be key to tasks such as object classification or detection (Chatoux, 2016). These feature descriptors could then be used as inputs to two systems: visual bags of words and histogram of features. We chose to use those systems because both of their implementations included a step that involved converting an image's feature descriptors to a standardized input. We found this necessary because different images can produce different numbers of feature descriptors. Each image had different dimensions and the feature detectors collected a different number of keypoints for each image. As a result, standardizing the input size of an image's feature descriptors was critical to creating a system that can use any arbitrary image.

After using the dataset to train and optimize all three of these systems, we tested how their accuracy metrics changed when predicting whether cars were in a set of augmented images. We produced this set of images firstly by randomly selecting 1600 images from our dataset, 800 from each class. Then, for each selected image, we applied each of the following augmentations, creating a new image after each augmentation.

- Increasing brightness by 1.75 times.
- Horizontally flipping the image.
- Zooming into the image by 1.25 times.
- Rotating the image by 0 to 30 degrees.
- Combining zoom by 1.25 times, rotation (0-30 degrees), and brightness by 1.75 times.

The accuracy metrics that we used to evaluate our car recognition systems were precision, recall, F1 score, and ROC AUC score. We chose these to make it easier to find details regarding our systems' flaws. We also used these metrics to find false positives or false negatives in our classification. We also used the ROC AUC score because it gives a measure of the overall accuracy of a given classification system. We used these scores to collect each system's accuracy metrics when testing it with unaugmented and augmented images. Then we compared the accuracy metrics of visual bags of words and histograms of features to the metrics obtained from AlexNet.

We will now describe the structure of our dataset in detail and the classification systems that we used.

4.1: Dataset

We obtained our cars dataset by using images from the 'Stanford Cars Dataset'(Krause, 2013). We chose this dataset because it has a large number of images containing cars in a variety of settings and arbitrary orientations, which is desirable for training machine learning algorithms. The Stanford Cars Dataset consists of 8143 images of cars of various classes in different orientations and lighting conditions. **Figure 3** shows some examples of car images that were collected.

Figure 3: Examples of car images that were collected.

We obtained our non-car images from the 'Open Images Dataset' (Kuznetsova, 2020). This dataset was chosen because it has a variety of images from numerous classes that can be easily obtained. We obtained roughly 175 images each from 50 different classes. The classes were chosen randomly and the images were aggregated to form the non-car images dataset for our project. **Figure 4** shows some examples of non-car images that were collected.

Figure 4: Examples of non-car images that were collected.

We intended to keep the ratio of car images to non-car images to roughly 1:1. This was done to avoid any learning bias towards any particular class that would potentially skew the results. The final split of our dataset was 8193 cars: 8470 non-car images. The images were further split into training, validation and test sets with the split ratio of 8:1:1. We used metrics from validation sets to evaluate the quality of our systems. We used images from the training set to train our systems. We used metrics from the test sets to estimate how well our systems would handle images outside of the training dataset.

After the split, the final dataset was as follows:

	Car Images	Non-car Images
Training (80%)	6554	6776
Testing (10%)	820	847
Validation (10%)	819	847

Figure 5: A table displaying the class distribution of dataset images.

Before using the images in the dataset for our systems, we resized the images to 256x256 pixels to reduce the computation time and also to standardize our input. The images were also converted to grayscale, since grayscale images only have 1 channel of pixel intensities instead of 3 channels in color images, resulting in faster computation.

4.2: Visual Bag of Words

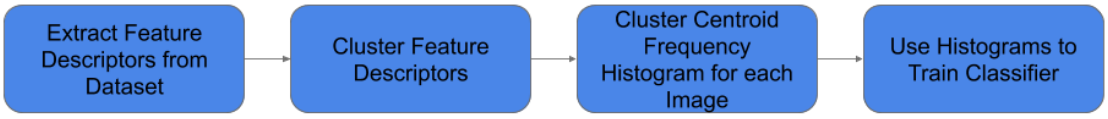

Figure 6: An illustration of a Visual Bag of Words system.

Visual bags of words systems are based on bags of words systems, which count certain words in documents and use those words' frequency as input for a machine learning classification algorithm. However, instead of counting words, visual bags of words systems

attempt to count the frequency of certain image features. (Csurka, 2004) Our visual bags of words system is based on the 2011 "**Object Classification and Localization Using SURF Descriptors**" paper. We chose to use its approach because it achieved an accuracy of over 90 percent (Schmitt, 2011).

Implementing a visual bag of words system is a multi-step process. The first step is to use a feature detector on every image in the dataset and gather all of the images' feature descriptors. We attempted to use visual bags of words with two types of feature descriptors: ORB and FREAK. We did this so we could choose the feature descriptor that resulted in better accuracy metrics. We did not use HOG with visual bags of words because it outputs a single global descriptor for each image. Because visual bags of words attempts to create a global descriptor for each image, using HOG would invalidate the need for visual bags of words.

The second step is to cluster all of the feature descriptors collected in the first step. Clustering is the process of separating data into groups so that similar data are in the same group. The purpose of clustering the feature descriptors is to determine which features are similar so that the visual bags of words system can identify such features in the future. We used K-means clustering to cluster the feature descriptors. K-means clustering is an unsupervised machine learning algorithm that carries out clustering. During training, it uses its input to calculate the locations of numerous cluster centers. The centers are indexed, and each cluster's index is used to label data inside of it. One can provide input data to a trained K-means clustering model to obtain the data's corresponding cluster labels. For each feature descriptor that we used in visual bags of words, we measured its accuracy metrics when using 300, 500, 800, 1000, 1500, and 2000 clusters.

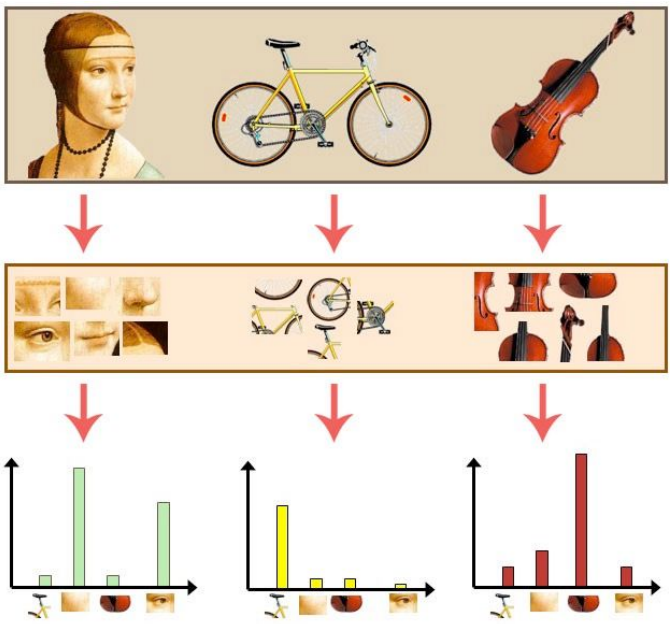

Figure 7: An illustration of the first three steps of implementing visual bags of words system (Fei-Fei, 2009).

The third step is to convert each image's feature descriptors to cluster labels by inputting them into the K-means clustering model that was trained in the second step. After obtaining the cluster labels, we count the number of times each cluster label appears in the image to create a frequency histogram of cluster labels. The final step is to use the images' frequency histograms as inputs for a machine learning classification algorithm. We used a support vector machine classifier for this purpose, which the "Object Classification and Localization Using SURF Descriptors" paper also used. (Schmitt, 2011) The paper also attempted to use a Naive Bayes algorithm, but time constraints prevented us from testing that algorithm with our system.

4.3: Histogram of Features

Histograms of features is a system based on combined frequency histograms of image patch feature descriptions. This method creates a uniform feature description for each image, making it suitable as an input for a machine learning model. Finding an image description using histograms of image patches can theoretically describe the entire image without much information loss. The feature descriptors used for this method were FREAK, ORB, and HOG. Our approach is based on a thesis (Sharma, 2014) that successfully identified faces with an average accuracy of over 96%.

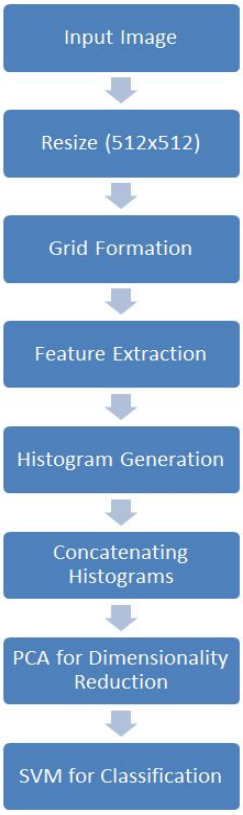

Figure 8: An illustration of a Histogram of Features system.

We used this method to standardize a feature descriptor shape for each image, making it easier to input into a machine learning algorithm. First, the image is resized to decrease computation time. After some experiments, we decided to use 512x512 images as this particular resolution could form a uniform (nxn) grid having square grid cells. The 512x512 images were then divided into square grids of 64x64 pixels each resulting in 8x8 (64) grids in total for each image. After the grids were formed, each grid cell's centroid was assigned to be a keypoint which would then be used for feature description. Since FREAK and ORB were local descriptors, they were used to describe each grid cell. We use each image patch's feature descriptor to form an intensity histogram that describes that particular image patch. All 64 histograms of a single image are then concatenated to form a uniformly shaped image description. Since images were divided into 64 cells and each grid would have an intensity histogram description of 256 bins, the resulting image description was of the shape (1, 16384).

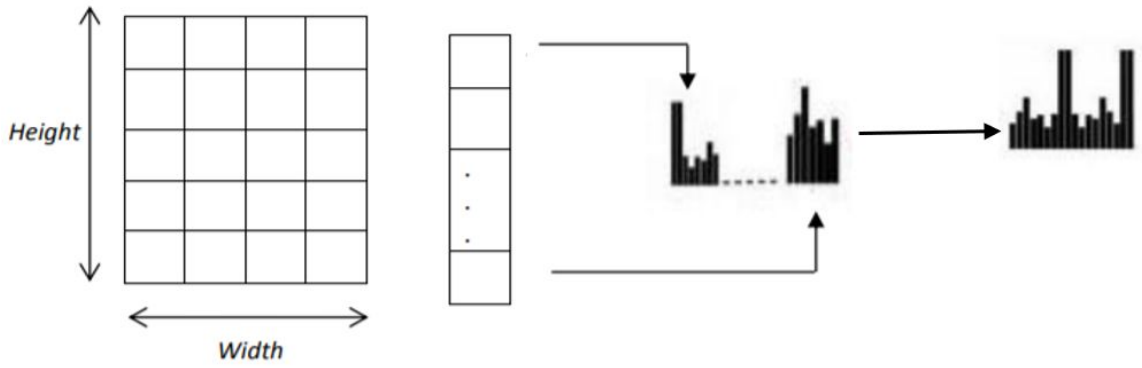

Figure 9: An illustration showing the histogram extraction process in the Histogram of Features system (Sharma, 2014).

To further reduce dimensionality and reduce computation time, the image description was run through a Principal Component Analysis (PCA). PCA essentially reduces the size of the image description to a desired number of components. The image descriptors are ordered based on their influence on the overall image description and the image descriptors within a certain variance range were selected. We tested this implementation for 500, 800, 1000, and 5000 components. The resulting image description would be of the shape (1,500), (1,800), (1,1000), and (1,5000) respectively.

Finally, the resulting image descriptions would be used to train a support vector machine(SVM), which is a machine learning algorithm for object classification. After labeling the training images appropriately based on their class (car vs non-car), the SVM is used to classify the images as car or non-car images. The SVM would then learn from labeled image descriptions and predict when new image descriptions are inputted to the system based on feature similarities.

4.3.1 Histogram of Global Descriptor:

Histogram of Gradients(HOG) is a global descriptor that is used to describe an image based on edge orientations. It is an effective method of describing an image since it preserves edge information very well. For an image of size 512x512, the HOG feature descriptor shape for an image is (1, 8192).

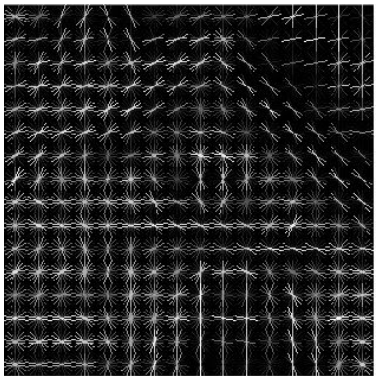

Figure 10: HoG Visualization for a random image (VLFeat, n.d.)

The dimensionality of the feature descriptor was further reduced to the size of (1,500) to reduce computation time. The resultant feature description after PCA is used to train the machine learning classifier which in this case was an SVM. The paper we based our approach on also attempted to use a Naive Bayes algorithm, but time constraints prevented us from testing that algorithm.

4.4. Convolutional Neural Network:

The primary reason we implemented a version of a convolutional neural network was to compare the effectiveness of hand-crafted feature descriptors to the performance of the learned features from convolutional neural networks. We also wished to understand the generalizability of the "crafted" and "learned" features when testing them with augmented images. For the purpose of comparison, our team decided to implement AlexNet (Krizhevsky, 2012) from scratch instead of using a pre-trained model. We did this to interpret the generalizability of a particular type of object's feature descriptors.

With respect to the implementation of AlexNet (Krizhevsky, 2012), there are a total of 11 layers. Typically, the convolutional layers are a variety of convolutional layers with different filters & window sizes, pooling, and batch normalization. After several convolutional layers, there is a flatten layer that converts the two-dimensional matrix into a one-dimensional array to be used as an input for fully connected layers used for classification. To implement a more scalable model given our resources, we chose to use three convolutional layers of different filter sizes - 96 -> 256 -> 128 instead of the filters used in AlexNet. The fully connected layers were

the same as the ones in AlexNet, and the output layer had two nodes in order to get a prediction probability for both classes given an image.

5: Results

In this section, we will outline all data collected from our experiments and how it lead to the parameters we chose for each of our systems.

5.1: Visual Bags of Words Results

Number of clusters	300	500	800	1000	1500	2000
Accuracy	0.695	0.760	0.710	0.726	0.742	0.713
F1	0.733	0.717	0.659	0.687	0.697	0.658
Precision	0.797	0.855	0.783	0.787	0.827	0.799
Recall	0.617	0.618	0.569	0.609	0.602	0.559
ROC AUC Score	0.816	0.871	0.811	0.828	0.850	0.822

Figure 11: Data collected when testing FREAK visual bags of words with different cluster numbers.

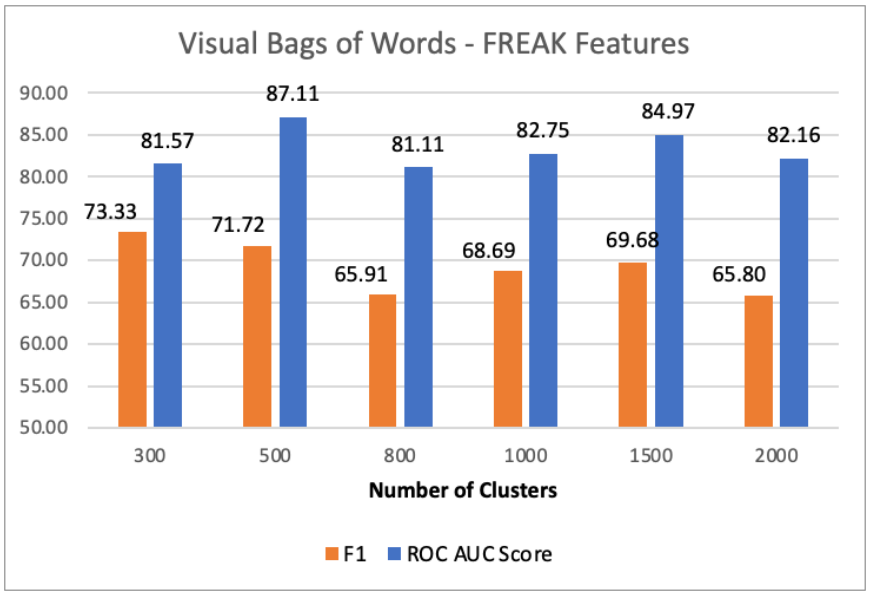

Figure 12: A chart showing the ROC AUC scores and F1 scores for FREAK visual bags of words when different numbers of clusters are used.

Number of Clusters	300	500	800	1000	1500	2000
Accuracy	0.794	0.815	0.803	0.814	0.833	0.824
F1	0.790	0.808	0.796	0.808	0.825	0.814
Precision	0.811	0.823	0.811	0.820	0.848	0.845
Recall	0.770	0.794	0.782	0.796	0.804	0.785
ROC AUC Score	0.798	0.814	0.803	0.814	0.832	0.823

Figure 13: Data collected when testing ORB visual bags of words with different cluster numbers.

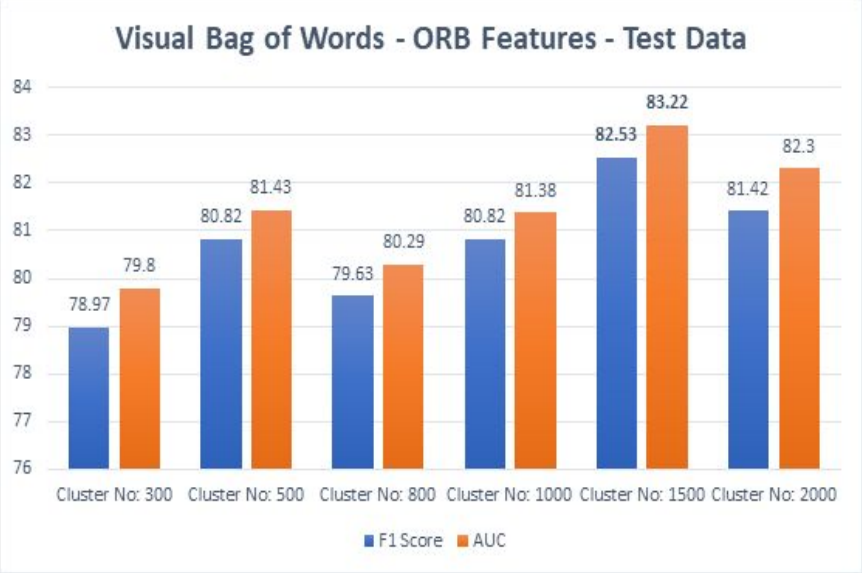

Figure 14: A chart showing the ROC AUC scores and F1 scores for ORB visual bags of words when different numbers of clusters are used.

While evaluating our visual bags of words systems, we attempted to find the number of clusters that lead to the best accuracy metrics for each feature descriptor used. **Figures 11-14** show how F1 score and ROC AUC score changed with the number of clusters when FREAK and ORB feature descriptors were used. We used the F1 score and ROC AUC score to measure the quality of the classifiers for two reasons. Firstly, we used F1 scores because they are higher when the classifier produces fewer false positives or false negatives overall. This allows us to tell whether our classifier produces many false positives or false negatives. Secondly, we used ROC AUC scores because they give a measure of accuracy by measuring the ability of the classifier to make correct predictions. As shown in **Figures 12 and 14**, visual bags of words gave the best

accuracy metrics for FREAK when it used 500 clusters. For ORB, 1500 clusters gave the best accuracy metrics.

5.2: Histogram of Features Results

HOG Metrics for 500 Components	Result
Accuracy	0.959
Precision	0.958
F1 Score	0.958
Recall	0.958
ROC AUC	0.959

Figure 15: A table showing the accuracy metrics obtained from using HOG features with 500 PCA components.

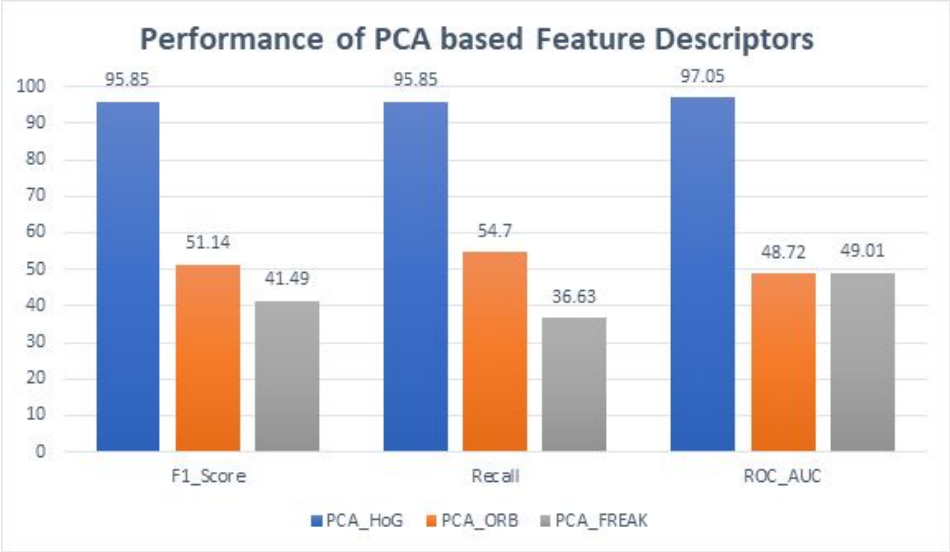

Figure 16: A diagram comparing the accuracy metrics of histogram of features systems with 500 PCA components using HOG, ORB, and FREAK.

When testing the histograms of features systems, using 500 PCA components resulted in the best accuracy metrics. Systems using other numbers of components resulted in accuracy metrics below 50%, and weren't considered for comparison as a result. **Figure 16** shows the systems' accuracy metrics when using 500 PCA components. Using FREAK and ORB resulted in accuracy metrics less than 50%. The histogram of features system using HOG vastly

outperformed the other two, with accuracy metrics of 96% on average. Its performance was approximately 92% better than that of the other two systems.

5.3 AlexNet Results

AlexNet Metrics	Result
Accuracy	0.962
Precision	0.971
F1 Score	0.961
Recall	0.951
ROC AUC	0.991

Figure 17: A table showing the accuracy metrics obtained from using AlexNet.

As shown in **Figure 17**, AlexNet produced relatively high metrics, with all of its metrics above 95%.

5.4. Data Augmentation Results

5.4.1. Visual Bag of Words

Augmentations	No Augmentation	Zoom	Rotated	Flip	Brightness change	Mixed
Accuracy	0.760	0.500	0.577	0.576	0.576	0.500
F1	0.717	0.200	0.178	0.215	0.215	0.184
Precision	0.855	0.500	0.654	0.700	0.700	0.500
Recall	0.618	0.125	0.103	0.127	0.127	0.112
ROC AUC Score	0.871	0.500	0.533	0.536	0.536	0.500

Figure 18: Data collected when testing FREAK visual bags of words against augmented images.

Augmentations	No Augmentation	Zoom	Rotated	Flip	Brightness change	Mixed
Accuracy	0.833	0.500	0.846	0.816	0.500	0.500
F1	0.825	0.606	0.842	0.809	0.627	0.227
Precision	0.848	0.500	0.850	0.834	0.500	0.500
Recall	0.804	0.768	0.833	0.784	0.842	0.147
ROC AUC Score	0.832	0.500	0.846	0.816	0.500	0.500

Figure 19: Data collected when testing ORB visual bags of words against augmented images.

After finding the number of clusters for each feature descriptor that produced the best accuracy metrics, we tested our visual bags of words systems with augmented images. **Figures 20-22** display how the accuracy metrics changed in response to various augmentations.

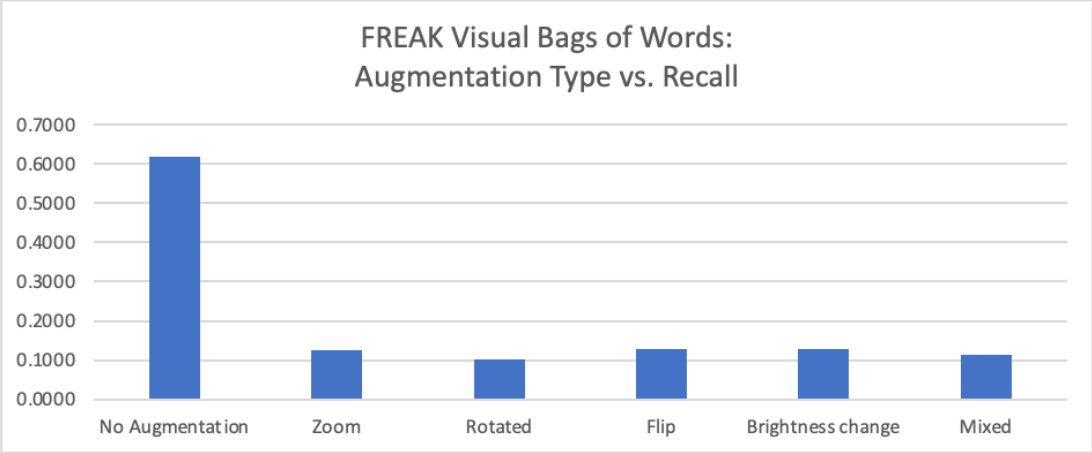

Figure 20: A chart showing the cross-validation recall of FREAK visual bags of words when tested with augmented images.

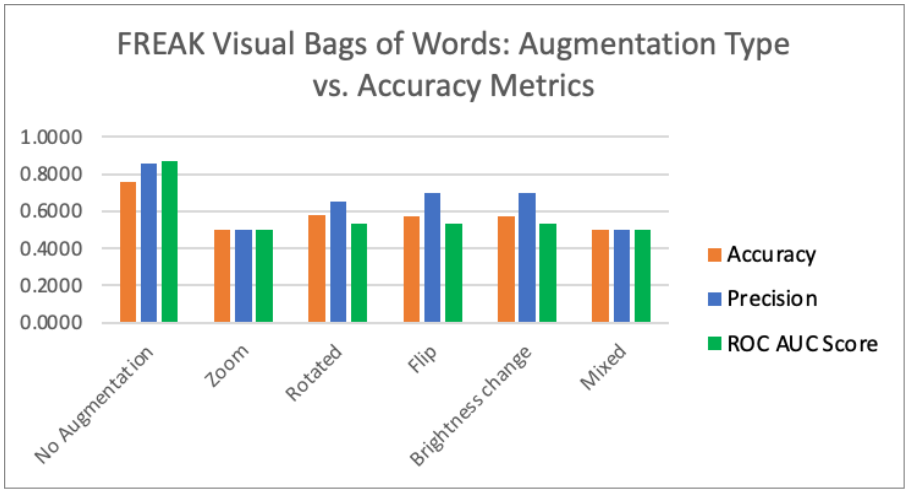

Figure 21: A chart showing the accuracy, precision, and ROC AUC score of FREAK visual bags of words when tested with augmented images.

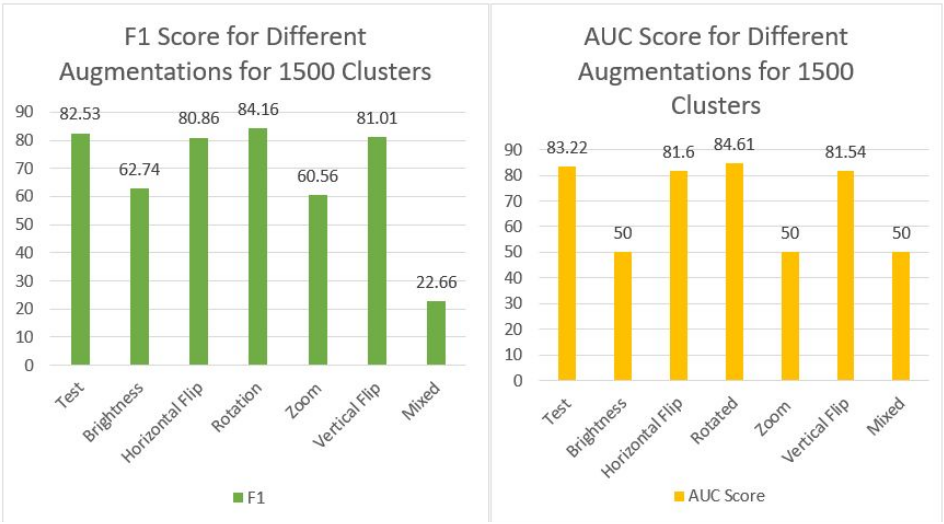

Figure 22: A chart showing the F1 score and ROC AUC score of ORB visual bags of words when tested with augmented images.

We found that when using FREAK feature descriptors, there was a 42% decrease in precision, F1 score, and ROC AUC score on average when making predictions for augmented images. Recall declined more sharply, decreasing by 79% on average. This means that when using FREAK, a significant amount of predictions were false negatives. Therefore, the system was much less able to successfully detect cars in augmented images.

When using ORB feature descriptors, visual bags of words only suffered a decrease in accuracy when exposed to specific augmentations. **Figure 22** shows that accuracy metrics only changed slightly when flipped or rotated images were used. Meanwhile, using images with brightness, zoom or mixed augmentations lead to a decrease in ROC AUC score by approximately 38%. Brightness and zoom augmentations also lead to a 26% decrease in F1 score

on average, and mixed augmentations lead to an even steeper decline, causing a 75% decrease in F1 score.

5.4.2. Histogram of Features

HOG	Not Augmented	Augmented				
	Test	**Brightness**	**Zoom**	**Flipped**	**Rotation**	**Mixed**
Accuracy	0.959	0.971	0.917	0.967	0.876	0.696
Precision	0.958	0.972	0.970	0.972	0.968	0.934
F1 Score	0.958	0.969	0.911	0.966	0.862	0.557
Recall	0.958	0.967	0.859	0.959	0.778	0.397
ROC AUC	0.959	0.971	0.916	0.966	0.876	0.686

Figure 23: Data collected when testing HOG histogram of features with augmented images.

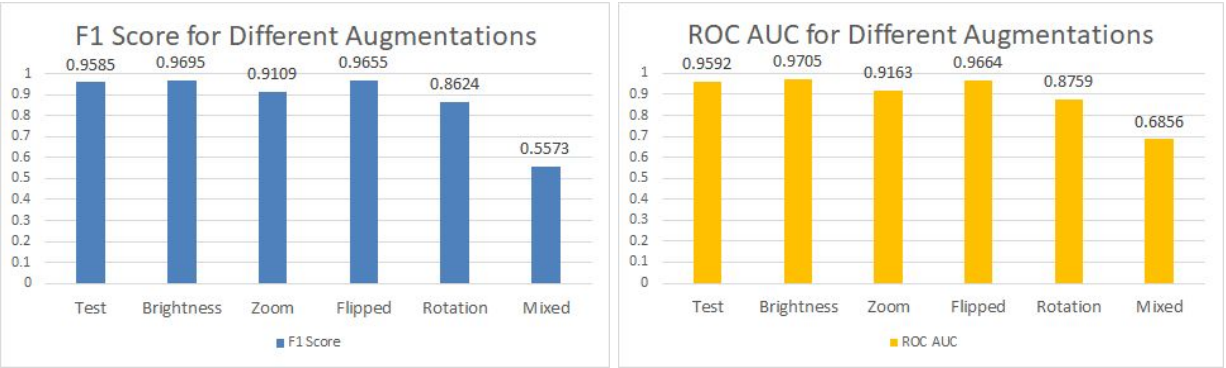

Figure 24: A chart showing the F1 scores and ROC AUC scores of HOG histogram of features when tested with augmented images.

We found that when using HoG features, performance metrics did not decrease much for brightness changes and flipped images. Whereas for zoomed and rotated images, the performance did drop by an average of 5-8% as shown in **Figure 23**. Mixed augmentation results were much more drastic with an average decrease in the performance metrics by about 35-40%. Therefore, the system worked well for slight variations from the training dataset but did not perform well when the images were highly augmented and there was a large variation of images from the training dataset.

5.4.3. AlexNet Implementation

AlexNet	Not Augmented	Augmented				
	Test	**Brightness**	**Zoom**	**Flipped**	**Rotation**	**Mixed**
Accuracy	0.962	0.979	0.897	0.962	0.871	0.687
Precision	0.971	0.991	0.962	0.967	0.974	0.973
F1 Score	0.961	0.978	0.889	0.965	0.855	0.539
Recall	0.951	0.965	0.826	0.963	0.762	0.373
ROC AUC	0.991	0.996	0.974	0.994	0.958	0.821

Figure 25: Data collected when testing Alexnet with augmented images.

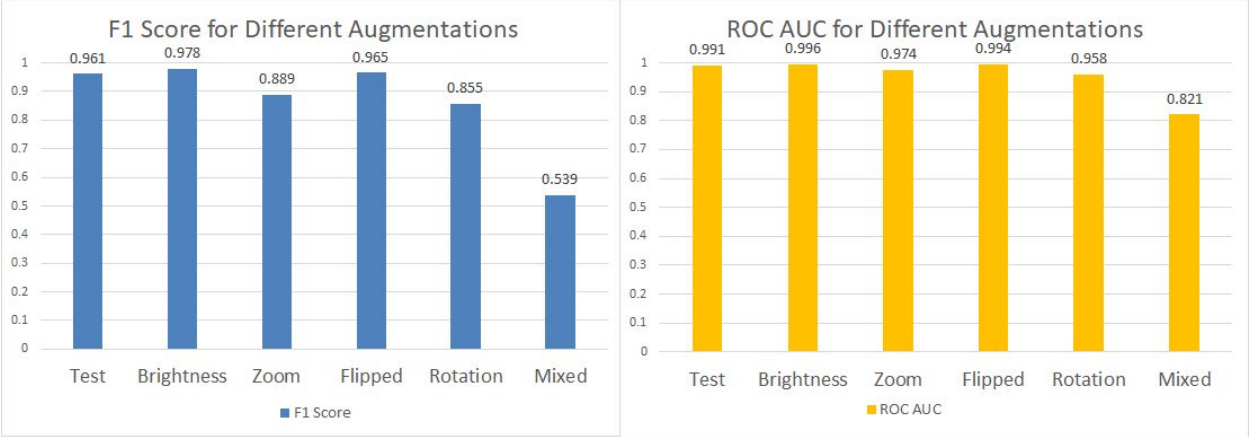

Figure 26: A chart showing the F1 score and ROC AUC score of AlexNet when tested with augmented images.

When AlexNet was used to predict the class of augmented images, its accuracy varied depending on the type of augmentation it was exposed to. For example, the average accuracy of AlexNet either improved or didn't decrease much when it was exposed to flipped images or images where brightness was altered. However, the performance dropped between 7-9% for images that were either scaled or rotated. Hence, we can conclude that AlexNet is less robust to scaling or rotation when it is trained on clean images.

6: Conclusions

This section will outline the results of our comparison between traditional computer vision systems and AlexNet in terms of accuracy metrics.

6.1. Accuracy Metrics Comparison

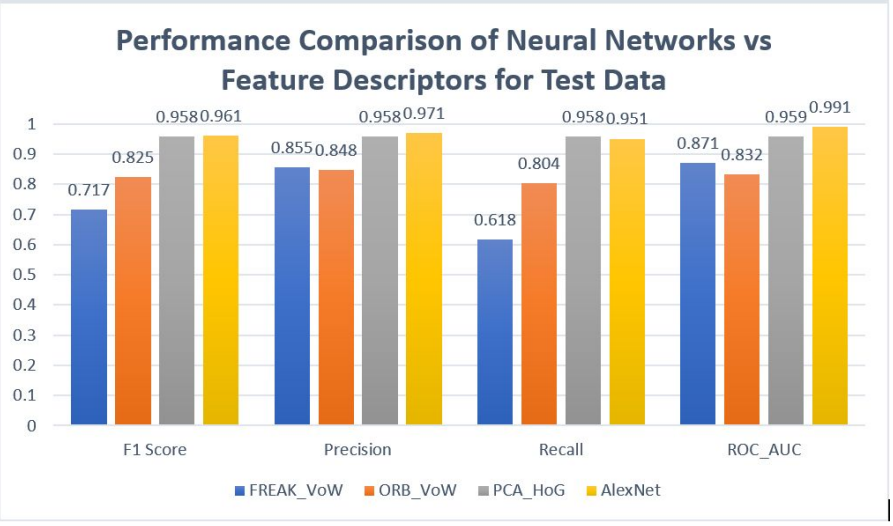

Figure 27: A diagram comparing the accuracy metrics of all implemented systems.

As **Figure 27** shows, AlexNet's accuracy metrics were higher than other systems' metrics on average, meaning it mostly outperformed the traditional computer vision systems. The recall and AUC score for AlexNet was much higher than visual bags of words (15% and 16% better on average). From this, we can conclude that in general, AlexNet was more accurate than traditional computer vision methods. Interestingly, HOG histogram of features had accuracy metrics close to that of AlexNet's and managed to produce a recall of 95.85, which was higher than AlexNet's recall of 95.1. Therefore, we can also conclude that a histogram of features using HOG descriptors is comparable to AlexNet in terms of accuracy.

6.2. Different Augmentation Comparison for Different Implementation

Figure 28: Comparison of performance of different implementations for different argumentation types.

As shown in **Figure 28**, AlexNet's accuracy metrics tended to outperform other systems' metrics when exposed to images with brightness, flipping, rotation, and zoom augmentations. In addition, a histogram of features based system using HOG was able to produce accuracy metrics comparable to AlexNet's when exposed to the same augmentations. From this, we can conclude that AlexNet is more robust to brightness, flipping, rotation and zoom augmentations than traditional computer vision systems. We can also conclude that a histogram of features based system using HOG is as robust to these augmentations as AlexNet.

6.3. Mixed Augmentation Comparison of Different Implementations

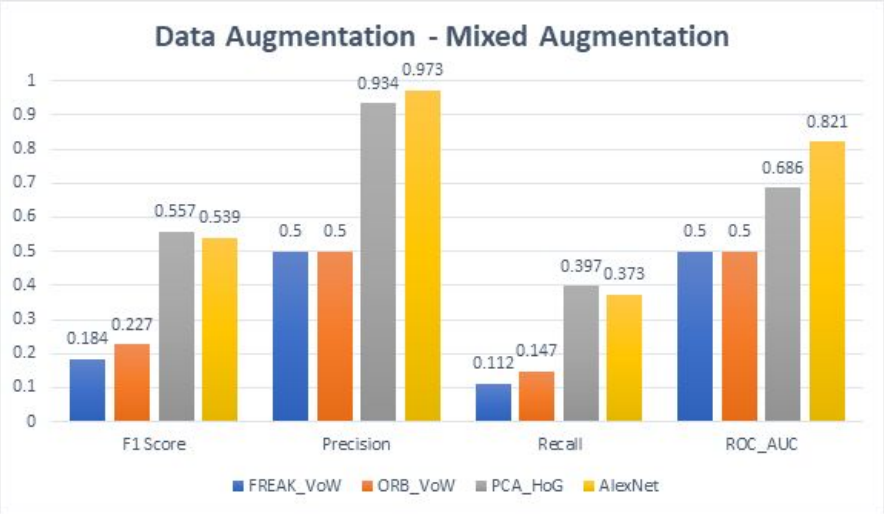

Figure 29: A chart comparing accuracy metrics for the implementations when tested with mixed augmentation images.

When the different systems were exposed to mixed augmentation (meaning each image had a fixed combination of brightness, zoom, flip, and rotation applied), the performance sharply decreased for all implementations. Visual bags of words using ORB descriptors displayed the worst performance, with average accuracy dropping to 22.6%. This was about 73% lower than the unaugmented test accuracy. The drop in accuracy metrics for histogram of features and AlexNet was by 42% and 28%, respectively. AlexNet performed significantly better than the Histogram of Features method, but both algorithms achieved an average accuracy of less than 70%.

From our results, some conclusions that can be made are:

1. AlexNet is more accurate and more robust to augmented images when compared to systems using traditional computer vision methods.
2. The classification accuracy depends primarily on the “learned” features from familiar images or feature descriptors. If recognition models don’t learn certain features, they are less likely to correctly classify those features.
3. This investigation can be further extended by studying how to improve the performance and the generalizability of these implementations when tested with augmented images without being exposed or trained with augmented images.
4. This investigation can also be extended for images when a car is present in a cluttered environment along with other objects and test the performance metrics for such conditions.

7: References

Chatoux, H., Lecellier, F., & Fernandez-Maloigne, C. (2016, 2016-12-04). Comparative Study of Descriptors with Dense Key points. Paper presented at the 23rd International Conference on Pattern Recognition, Cancun, Mexico.

Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004). Visual categorization with bags of keypoints. Work Stat Learn Comput Vision, ECCV, Vol. 1.

Fei-Fei, L., Fergus, R., & Torralba, A. (2009). Recognizing and Learning Object Categories. In.

ImageNet. (2012). ImageNet Large Scale Visual Recognition Challenge 2012.

Krause, J., Stark, M., Deng, J., & Fei-Fei, L. (2013, 2-8 Dec. 2013). 3D Object Representations for Fine-Grained Categorization. Paper presented at the 2013 IEEE International Conference on Computer Vision Workshops.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Paper presented at the Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, Lake Tahoe, Nevada.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., . . . Ferrari, V. (2020). The Open Images Dataset V4. International Journal of Computer Vision, 128(7), 1956-1981. doi:10.1007/s11263-020-01316-z

Schmitt, D., & Mccoy, N. (2011). Object Classification and Localization Using SURF Descriptors.

Sharma, R. (2014). Object Detection using Dimensionality Reduction on Image Descriptors. (Computer Engineering). Rochester Institute of Technology, KGCOE.

VLFeat. (n.d.). HOG features. Retrieved from https://www.vlfeat.org/overview/hog.html

8: List of Figures

9: Appendix

9.1: Project Code

All code used during our project can be found at the following GitHub repository: https://github.com/japierreSWE/cv-project