



School of Computing and Information Technology

CSCI946: Big Data Analytics

Spring 2024

Submitted By: Masud Zaman Faruk

Introduction:

The primary objective of this assignment is to provide practical experience in Big Data Image Analysis. The Big Data Analytics Lifecycle will be followed, encompassing various steps to guide the experiment and facilitate the reporting of findings and conclusions. This project focuses on the ImageNet dataset, a renowned and widely utilized benchmark for training and evaluating models in image classification. ImageNet includes an extensive collection of labeled images across diverse categories, and models trained on this dataset typically demonstrate high accuracy when evaluated using ImageNet's standard validation sets.

However, a challenge arises when these models are tested on alternative validation sets not included in the original training data. In such cases, a significant drop in accuracy is often observed, indicating that the model's performance is less effective than anticipated. The aim of this assignment is to investigate the causes of this accuracy drop and explore methods to enhance the model's performance on new, unseen data. Factors such as data distribution, potential overfitting, and differences between the training and new datasets will be examined. By the conclusion of this assignment, the project aims to provide insights into improving the reliability and accuracy of image classification models, even when applied to varied image data.

About ImageNet Data Sets:

For many years, the original ImageNet validation set, known as test set 1, has been used as the main benchmark to check how well image recognition models work. Lots of models have achieved great accuracy on this set, but there is a worry that they might be overfitting, which means they might be getting too used to certain patterns or details in this specific test set. So, while they do well on test set 1, they might not do as well on new or different kinds of images.

To check if this is true, another test set, called test set 2, was created using the ImageNetV2 validation set. This new set was made using a method called

‘MatchedFrequency,’ which tries to pick images that are similar to the original ImageNet set but with fresh, new images. The goal of using test set 2 is to see how well the models can work with new data, not just the same data they’ve already seen before.

For this project, we were given deep image features that were taken from a large pre-trained model. These features come from the ImageNet training set, test set 1, and test set 2. By studying and comparing these sets, we hope to see how well the models perform across different data and figure out how to make them work better with new images.

Discovery

Problem Statement:

The data was loaded in chunks and shuffled to ensure random class distribution in a batch of 64 to avoid memory issue. The data is converted to TensorFlow Datasets and a validation split is created with 20% of the training data.

Prior to loading the data into TensorFlow Dataset, the unnecessary columns were discarded for analysis. The unnecessary column in our case is the “path” column which has links to the images, and the Unnamed index column.

Data Sources:

For this project, two well-known image datasets: ImageNet and ImageNetV2 are used. These are famous benchmarks that help to test how good computer vision models are. The ImageNet training set has over 1.2 million images that the models are trained on. The original ImageNet validation set, which has 50,000 images, is what we call test set 1. On the other hand, test set 2 is a smaller set of 10,000 images from ImageNetV2. It was made using a method called ‘MatchedFrequency,’ so it has similar types of images but with new ones.

Both datasets include 1,000 different types of images, showing lots of variety and real-world challenges that come with image analytics. We were given deep image features that were already extracted from a big pre-trained model, taken from the ImageNet training set, test set 1, and test set 2. There is also a practice task where we can try extracting features ourselves using a smaller pre-trained model on a smaller set of 10,000 images from ImageNetV2.

Overall, these well-known and complex datasets give us a chance to do a hands-on project in big data analytics. By working with them, we will learn how to handle large amounts of data, improve image classification models, and understand the challenges of working with different kinds of image data in the real world.

Data loading and preprocessing:

The data is loaded in chunks and shuffled to ensure random class distribution in a batch of 64 to avoid memory issue. The data is converted to TensorFlow Datasets and a validation split is created with 20% of the training data.

Prior to loading the data into TensorFlow Dataset, we discarded the unnecessary columns for our analysis. The unnecessary column in our case is the “path” column which has links to the images, and the Unnamed index column.

```
Feature shape: (64, 1024)
Label shape: (64,)
Labels: [774 187  20 736 326 753 902 742  90 567 331 702 507 789  17 560 324 933
 407 708 677  94 362 189 829 492 487 676 212 617 948 597 392 912 329 216
 115 664  68 412 542 716 141 119 109 960 758 355 580 512 837 940 213 868
 237 367 894 303 900 959   1 111 211 489]
```

Figure: Class distribution in a batch of 64

Class distribution of the training set is below

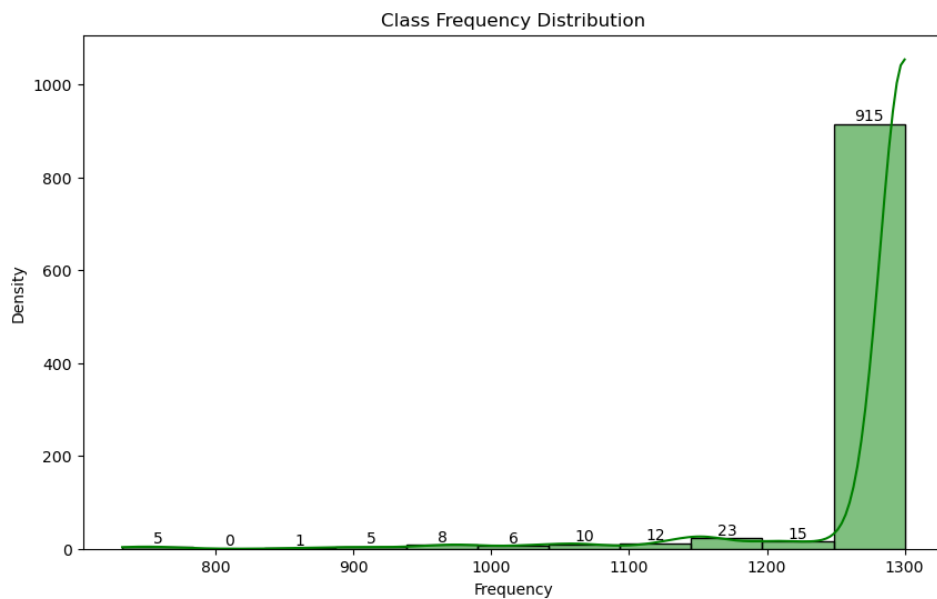


Figure: Sample distribution of the Classes

Model building:

For model selection, a Convolutional Neural Network (CNN) was chosen since the training data is extracted from image data. The model is created with 3 convolutional layers stacked on Dropout and Output layer. Dropout and learning rate parameter added to regulate overfitting.

I chose to go with a dropout of 0.5, learning rate of 0.001 (medium). The model has only been trained for 1 Epoch since the training process is computational and time intensive.

The model achieved a training accuracy of 88.16% and validation accuracy of 95%.

Training and Validation Accuracy and Loss:

accuracy: 0.8816 - loss: 0.6999 - val_accuracy: 0.9513 - val_loss: 0.1814

The model was tested on Test Set 1 and Test Set 2.

Model achieved 89% accuracy on Test Set 1 and 81% on Test Set 2. Which is a significant drop.

782/782 ————— 53s 66ms/step - accuracy: 0.8898 - loss: 0.5407
157/157 ————— 27s 175ms/step - accuracy: 0.8096 - loss: 1.0332

We can see that most classes in test set 1 were getting over 90% accuracy, but for test set 2, most of the classes received an accuracy of 70 – 80%

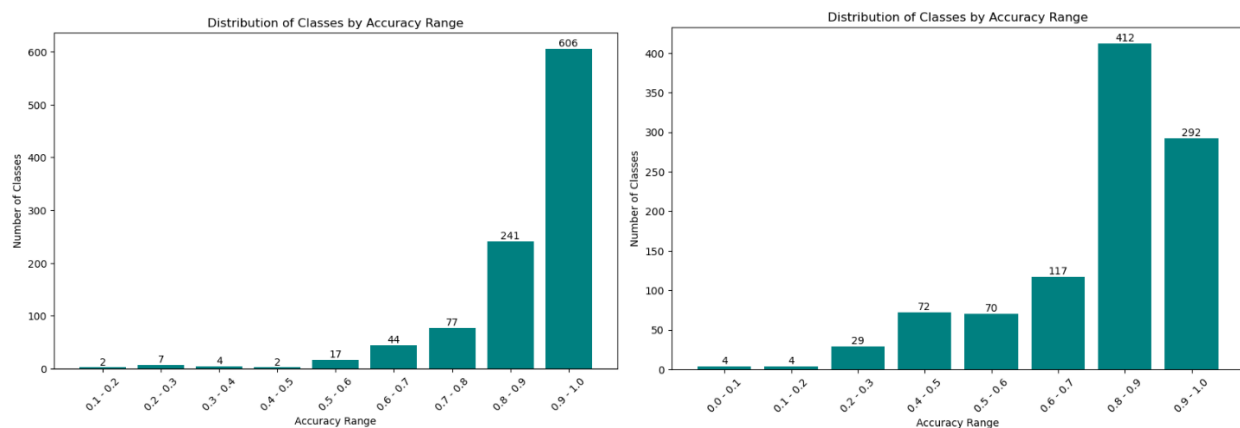


Figure: Class frequency in accuracy range

Analysis and Model finetuning:

Hypothesis testing: Performance drop on test set 2:

Hypothesis 1: Due to Imbalanced class distribution

The class distribution is investigated of both test sets to find whether all classes are distributed evenly in both train and test sets.

To test the hypothesis, the class frequency of the training set was plotted and found that out of 1000 classes, 915 have frequency of 1300 in the training set and no classes have frequency less than 732.

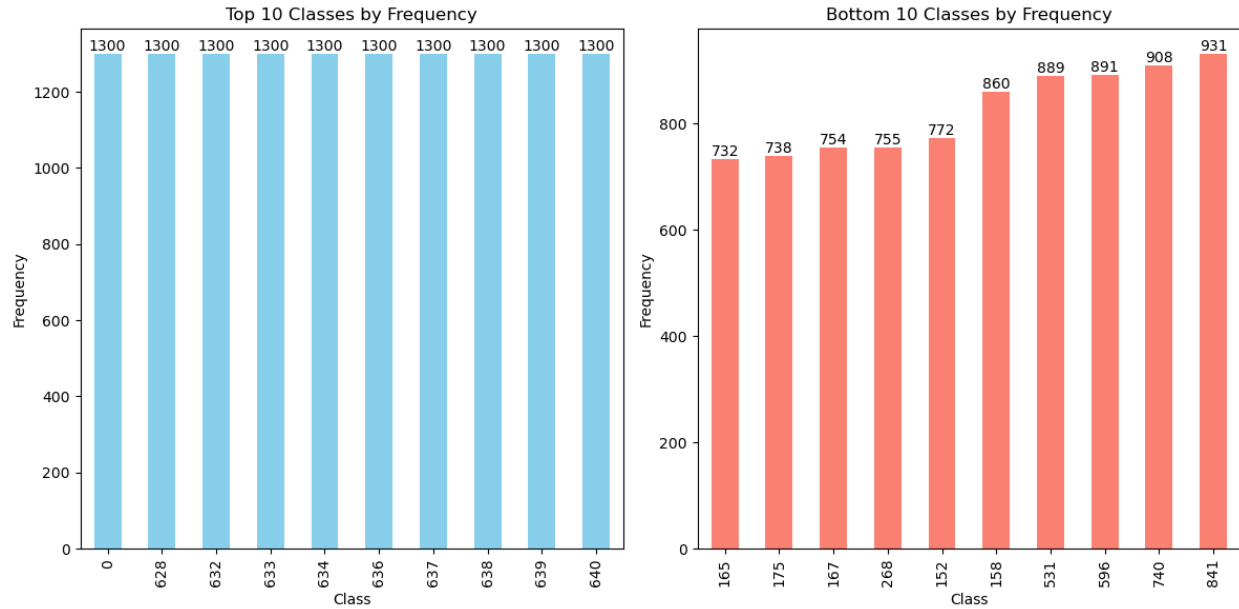


Figure: Class distributions in the training set

Subsequently, an investigation was conducted to determine whether the classes that exhibited poorer performance in test set 2 were affected by low class distribution in the training dataset. The analysis identified the following classes as having the most significant performance drop in test set 2.

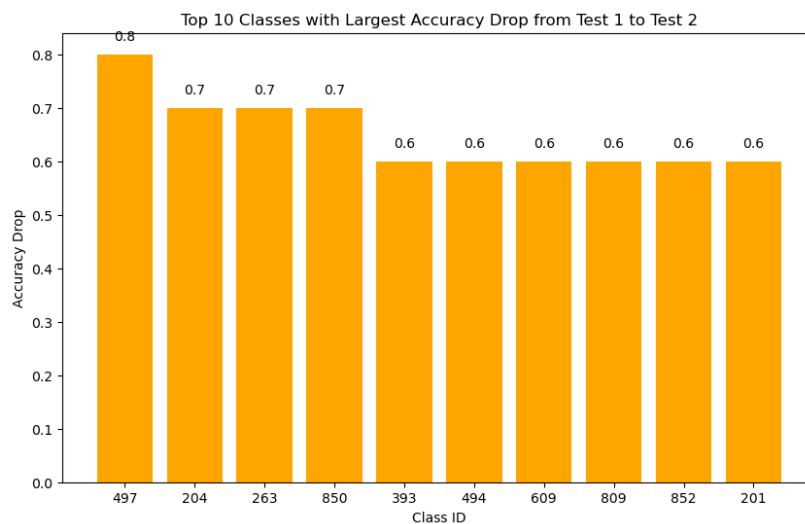


Figure: Classes with the biggest performance drop on Test Set 2

Upon analyzing the training data, it was observed that the top poorly performing classes in test set 2 actually have the highest class distribution in the training dataset.

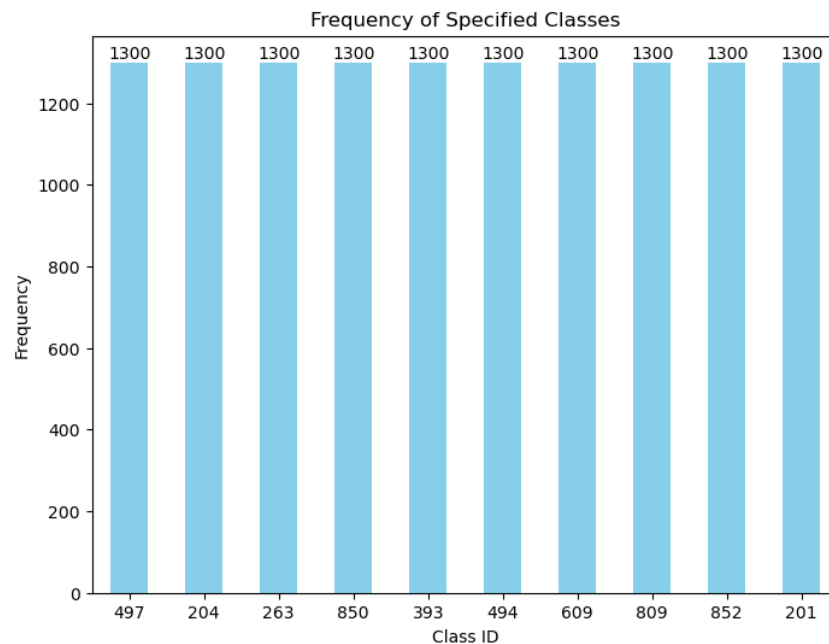


Figure: Class distribution of the bad performing classes in the training data

From this investigation, it can be concluded that test 2 performance drop is not due to the imbalanced class distribution.

Hypothesis 2: Due to overfitting

To investigate whether the model is performing worse on test set 2 due to overfitting or adapting to the training data, the model learning was reduced to very small (0.0001) so that it is easier for the optimizer to find global minima of the gradients. The model was then trained on test set 2.

After the experiment, we were able to improve the performance (accuracy) of the model on test set 2, but the performance gap between the 2 test sets remains significant $(89 - 82) = 7$.

782/782	—————	35s 45ms/step - accuracy: 0.8905 - loss: 0.5084
157/157	—————	8s 48ms/step - accuracy: 0.8169 - loss: 0.9612

It can be concluded that the model is not performing bad on test set 2 due to overfitting and adaptivity.

Further analysis of the feature space:

To examine feature distribution differences between Test Set 1 and Test Set 2, a t-test is conducted, identifying features with statistically significant variations between the two sets. Some features showed clear distribution differences, indicating that Test Set 2 has unique characteristics that may contribute to the model's performance drop.

This insight points to the need for further feature analysis and possibly targeted retraining to improve performance on more challenging images.

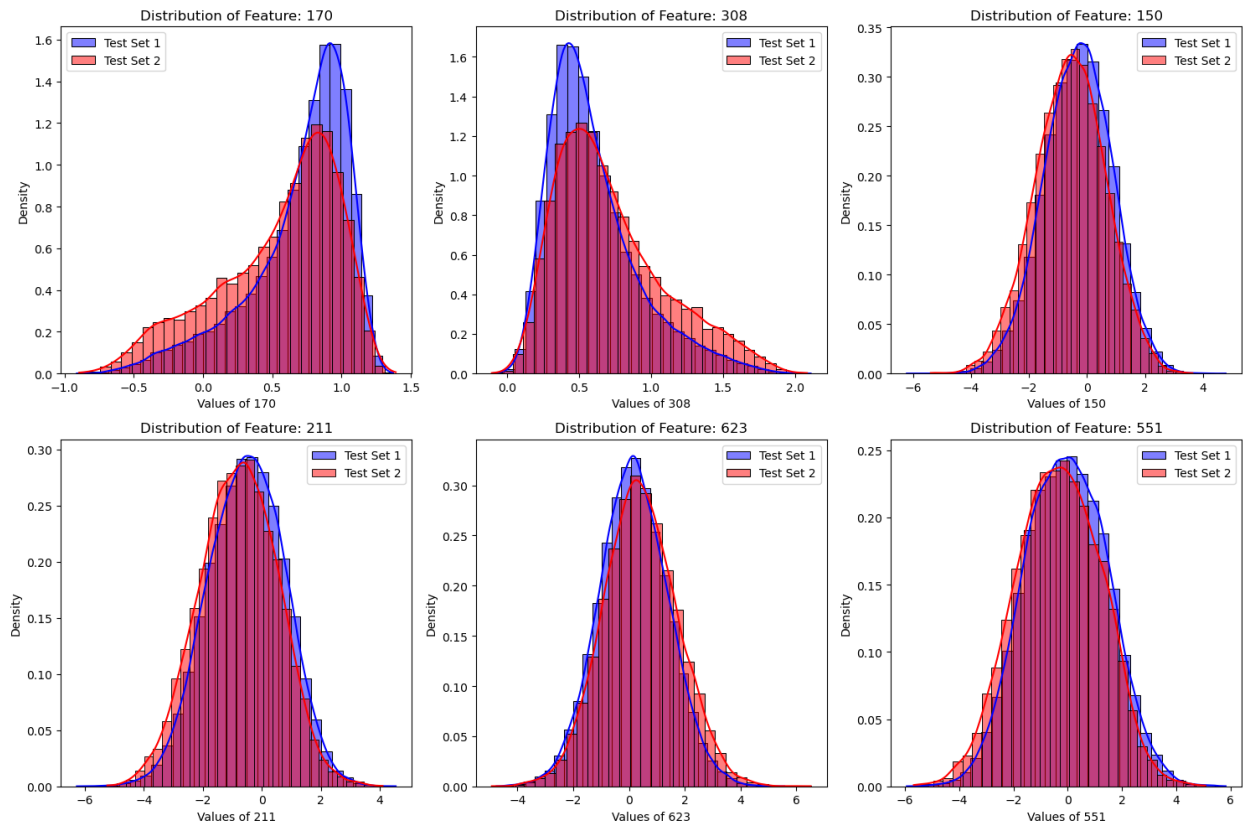


Figure: Feature value distribution of the significantly different features in test sets

To analyse further, statistical analysis was conducted to find which features have significantly different values when compared values in correct classification and misclassification for said specific feature.

Significant Features in Test Set 1:

	Feature	Mean_Correct	Mean_Misclassified	Mean_Diff	Variance_Correct	Variance_Misclassified	P_Value
879	879	-0.242073	-0.308433	0.066361	1.840498	1.865991	0.000776
260	260	0.197838	0.262247	-0.064409	2.184486	2.137011	0.002343
447	447	-0.110907	-0.167912	0.057006	1.989357	1.971382	0.005015
278	278	0.187707	0.128827	0.058880	2.124787	2.153319	0.005494
682	682	-0.227107	-0.282695	0.055588	2.094984	2.093515	0.007894
848	848	0.208495	0.263298	-0.054803	2.035526	2.063630	0.008291
93	93	0.146554	0.201764	-0.055210	2.118898	2.133910	0.008936
974	974	0.057784	0.108698	-0.050914	1.849011	1.886135	0.010279
489	489	0.000805	0.054434	-0.053630	2.076717	2.118194	0.010747
980	980	0.008440	0.061421	-0.052981	2.130545	2.159848	0.012603

Significant Features in Test Set 2:

	Feature	Mean_Correct	Mean_Misclassified	Mean_Diff	Variance_Correct	Variance_Misclassified	P_Value
397	397	0.280678	0.404256	-0.123578	2.024777	2.037918	0.000704
594	594	0.043952	-0.072434	0.116386	2.064458	2.095328	0.001633
981	981	-0.196497	-0.317800	0.121304	2.329028	2.283806	0.001722
516	516	-0.032862	-0.155193	0.122331	2.255710	2.370201	0.001791
381	381	-0.155467	-0.035739	-0.119728	2.270596	2.410772	0.002417
990	990	0.155575	0.034842	0.120733	2.432621	2.549234	0.002965
340	340	0.105864	0.209580	-0.103717	1.984531	1.892875	0.003323
806	806	0.190547	0.074275	0.116272	2.484643	2.465337	0.003789
765	765	0.160232	0.271574	-0.111342	2.224377	2.283818	0.003855
1014	1014	0.095080	-0.016260	0.111341	2.226255	2.299825	0.003960

A cluster analysis was done on mean differences of significant features (low P values) for both test sets. Later significant features of test set 1 and significant features of test set 2 were clustered along mean difference and variance difference.

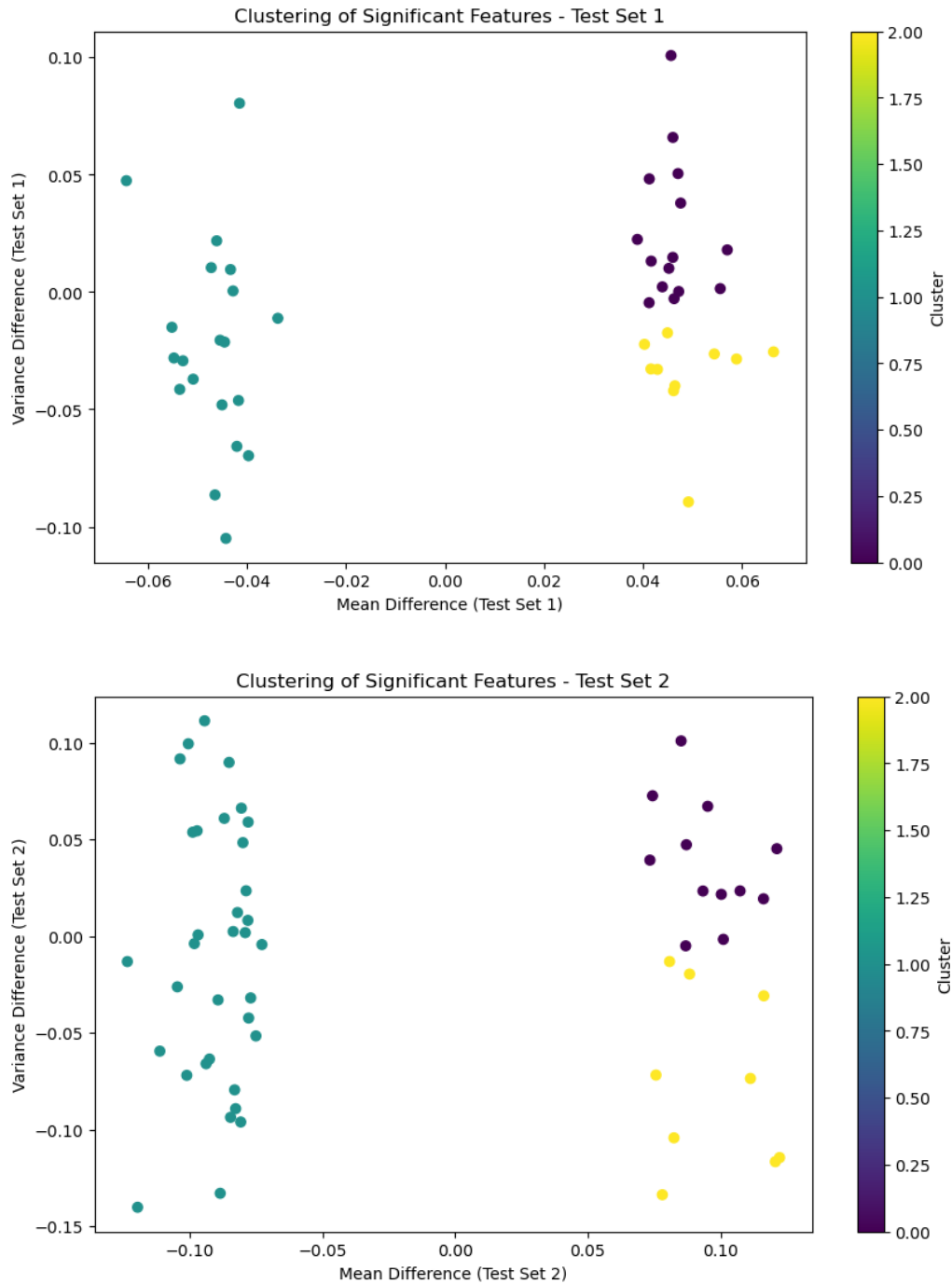


Figure: Mean difference vs Variance difference for the test sets

It is evident that the clusters are divided into 2 main clusters:

Yellow cluster have high positive mean difference and high negative variance difference.


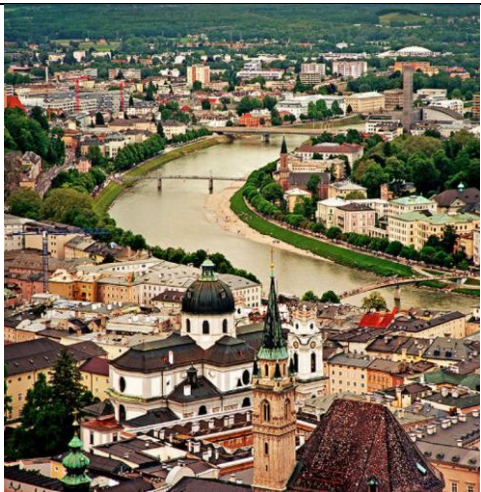


Which means high mean values for these features, the model was able to predict more accurately while variance was low.

For purple cluster, the features having high mean values and high variance helped the model predict better.

For green cluster, these features negative mean difference meaning reducing the feature value will enable the model to perform better.

To conclude, it can be stated that the model is performing bad on test set 2 because the test set 2 features are different (or “hard”). We can support our case further if we pick a badly classified class and compare the pictures in both test sets.

From the analysis of classes that experienced the most performance accuracy: the top 2 classes were picked to compare the pictures.

Class	Train Set	Test Set 2
497 – Church 80% drop	<div>13</div> 	
204 – Lhasa 70% drop		

<p>850 – Teddy 70% drop</p>		
-------------------------------------	---	--

Result and Discussion:

The model performed well on Test Set 1, showing it can generalize accurately on that data. However, its accuracy dropped noticeably on Test Set 2, revealing some challenges with this set that weren't as present in the first.

After careful analysis, it was found that this drop in performance isn't due to class imbalance or overfitting. The classes that had the biggest accuracy drop in Test Set 2 were well-represented in the training data, so there was no shortage of examples for these classes. Also, when the model's learning rate was adjusted to prevent overfitting, it didn't close the gap in performance between the two test sets, meaning the model isn't just "memorizing" the training data.

The findings indicate that Test Set 2 has images with more complex or difficult-to-recognize features, which make accurate predictions harder for the model. These images might have more variety in textures, lighting, angles, or other factors that add complexity. This variety seems to go beyond what the model learned from the training data, leading to misclassifications.

In summary, while the model performs well on simpler images (as seen in Test Set 1), Test Set 2 may require adjustments to the model or additional feature engineering to better handle the complex features. Addressing this complexity could help close the performance gap and improve the model's ability to generalize across different types of images.

References:

[1] B. Recht, L. Schmidt, U. Berkeley, and V. Shankar, et al. "Do ImageNet Classifiers Generalize to ImageNet?" arXiv preprint arXiv:1902.01024, 2019,

Available: <http://people.csail.mit.edu/ludwigs/papers/imagenet.pdf>