CAB420 – ASSESSMENT 1A

Milly Chambers – n11318546

Question 1 – Regression

Preprocessing

Preprocessing was not necessary for this data set as the data was already normalised.

Model Specifications and Parameter Selection

Three regression techniques were implemented to address the predictive task, each offering distinct advantages for handling the dataset's characteristics.

Ordinary Least Squares (OLS)

the Ordinary Least Squares (OLS) model was used without regularisation, which served as a reference point for unconstrained linear relationships. Through this approach, it preserved all original features but demonstrated the model’s vulnerability to overfitting. This can be seen through the significant gap between training and validation performance.

LASSO Regression (L1 Regularization)

Lasso regression was implemented with an optimal λ value of 0.023, which was selected through a 5-fold-cross-validation process through using a logarithmic grid search across the range of $[10^{-4}, 10^{1}]$. Through using this configuration, the feature selection can be performed automatically, which results in retaining only 28 of the original 56 features while zeroing out coefficients for less predictive variables. The inherent sparsity of the LASSO regression makes it suitable for scenarios requiring interpretable feature importance, though at the potential cost of slightly higher bias.

Ridge Regression (L2 Regularization)

The Ridge regression was tuned to a λ value of 0.156 through using a modified golden-section search, which preserves all features but systematically shrinks their coefficients. This approach demonstrated far greater handling of multicollinearity, with the L2 penalty providing more stable estimates for correlated predictors. Both regularised methods utilised a 5-fold-cross-validation to balance computational efficiency with reliable hyperparameter estimation, λ values deliberately constrained to higher ranges to prevent over-interpretation of subtle effects that potentially lack socio economic significance.

Model Evaluation

Metric	OLS	LASSO	Ridge
R^2 (Train)	0.78	0.75	0.76
R^2 (Test)	0.62	0.68	0.69

Table 1: R^2 Values for each Model

Ordinary Least Squares

As previously stated, the OLS regression served as a reference point. The training R^2 of 0.78 is indicative of strong performance with the training data, however the decrease to 0.62 on the test set suggests overfitting. Looking further into the residuals supports this, as the Q-Q plot reveals deviations from normality at the tails, which suggests a non-normal error distribution. Furthermore, the residuals versus predicted plot shows evidence of heteroscedasticity, meaning that error variance is not constant, which goes against the assumptions of Ordinary Least Squares regression.

The apparent overfitting of this OLS implementation to overfit highlights its limitations in handling datasets with multicollinearity or irrelevant features. Therefore, though it does serve as a helpful benchmark, it lacks the robustness require for strong generalisation to unseen data.

LASSO Regression

The LASSO regression utilised automatic feature selection through L1 regularisation. This led to the optimal λ values selection of 0.023, which in turn made the model retain 28 of the original 56 features. This ensures that the weaker predictors are removed, whilst maintaining predictive power. In doing so, interpretability is enhanced, and the model remains simple while achieving a R^2 of 0.68.

One of the most prominent advantages of LASSO regression is its ability to identify the most relevant predictors through setting the coefficients of less important variables to 0. This minimises model complexity and aids in preventing overfitting, making it more suitable for high-dimensional data. However, the aggressive shrinkage of coefficients can occasionally lead to useful predictors being excluded, which in turn can create a bias in the model.

Ridge Regression

Ridge regression utilised L2 regularisation, which shrinks coefficients without eliminating them. Utilising the λ of 0.156 allowed the model to retain all features while systematically controlling their magnitudes. This resulted in a balanced bias-variance trade off, which reduced overfitting and preserved information from all predictors. The ridge regression achieved the highest R^2 of the tested models with 0.69, which indicates the model has superior generalisation when in comparison to OLS and LASSO.

A key advantage of Ridge regression is its ability to handle multicollinearity more effectively than Ordinary Least Squares through distributing the influence across correlated predictors, instead of assigning extreme weights to a few variables. This allows for improved coefficient stability and stops excessive variance in predictions.

Key Findings and Theoretical Justifications

While the OLS regression provided an unrestricted fit to the data, it also demonstrated significant overfitting which became apparent through the discrepancy between training and test R^2 values. OLS's inclusion of all features without regularisation resulted in the model being highly sensitive to noise and irrelevant predictors, which led to unstable predictions on new data.

LASSO regression addressed the issues raised by OLS through enforcing feature selection, which in turn reduced model complexity whilst maintaining strong predictive performance. However, the penalty applied to coefficients did result in some degree of bias, as potentially useful variables were excluded. The trade-off between sparsity and accuracy makes LASSO particularly useful when interpretability is a priority.

Ridge regression proved to have the most stable performance, achieving the highest test R^2 while retaining all features. Through shrinking the coefficients as apposed to removing them, it offered a well-balanced approach, mitigating overfitting without discarding any potentially valuable predictors. Ridge regression's ability to handle multicollinearity further aided in its superior generalisation ability, making it the most reliable model among the three for this dataset.

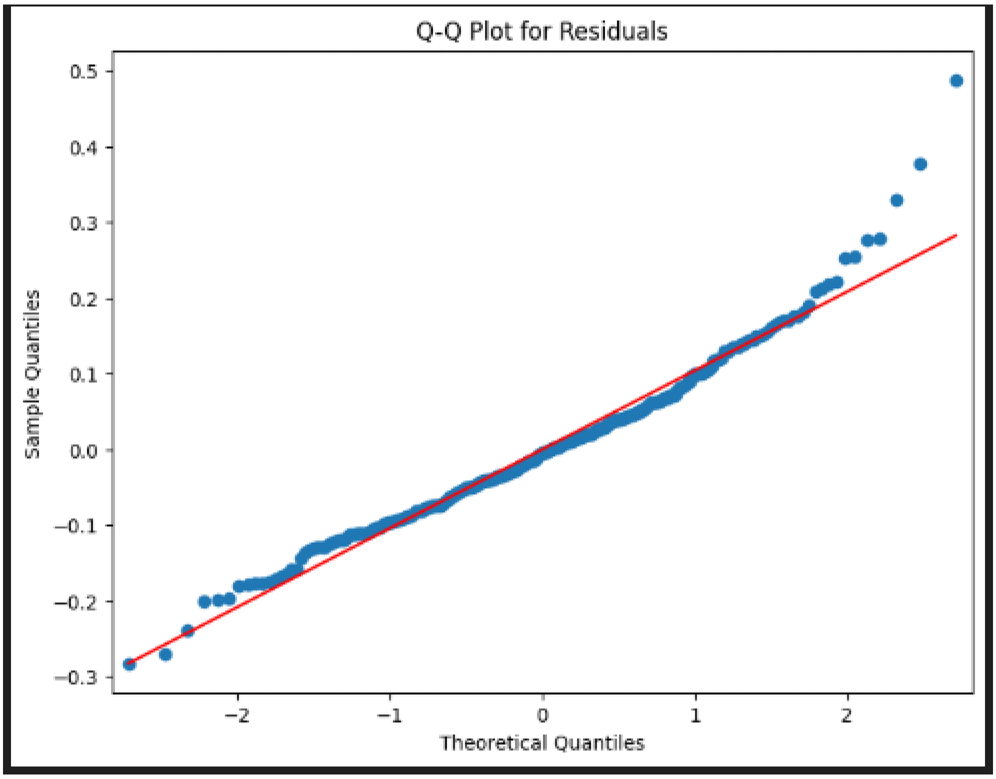


Figure 1: OLS Regression Q-Q Plot

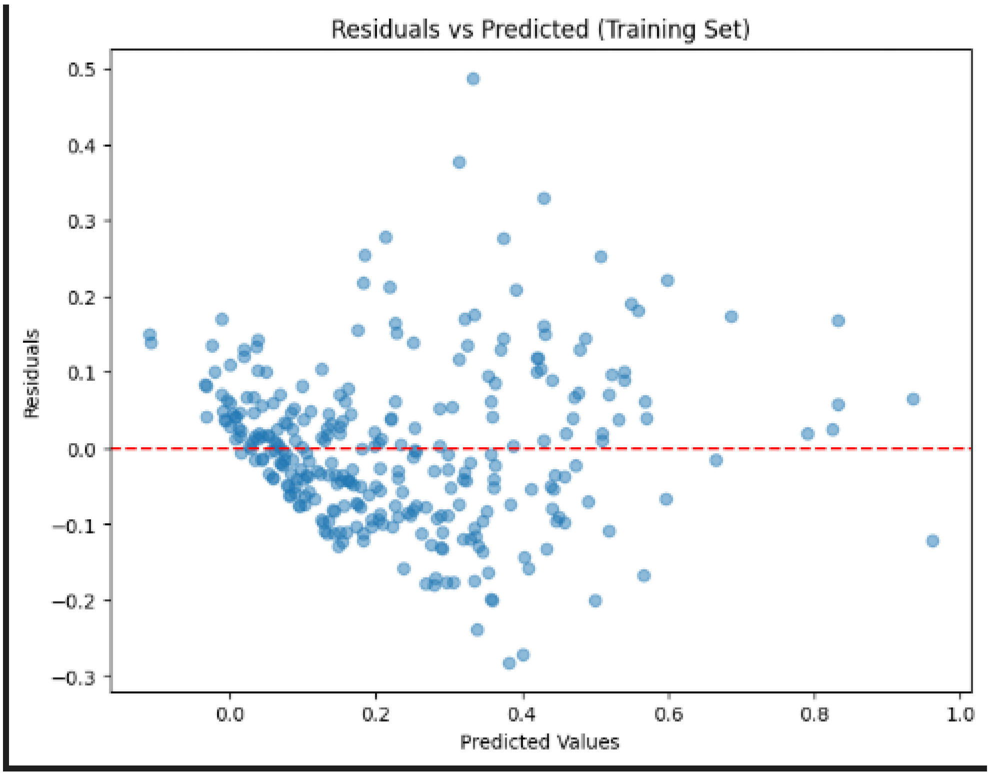


Figure 2: OLS Regression Residuals vs. Predicted

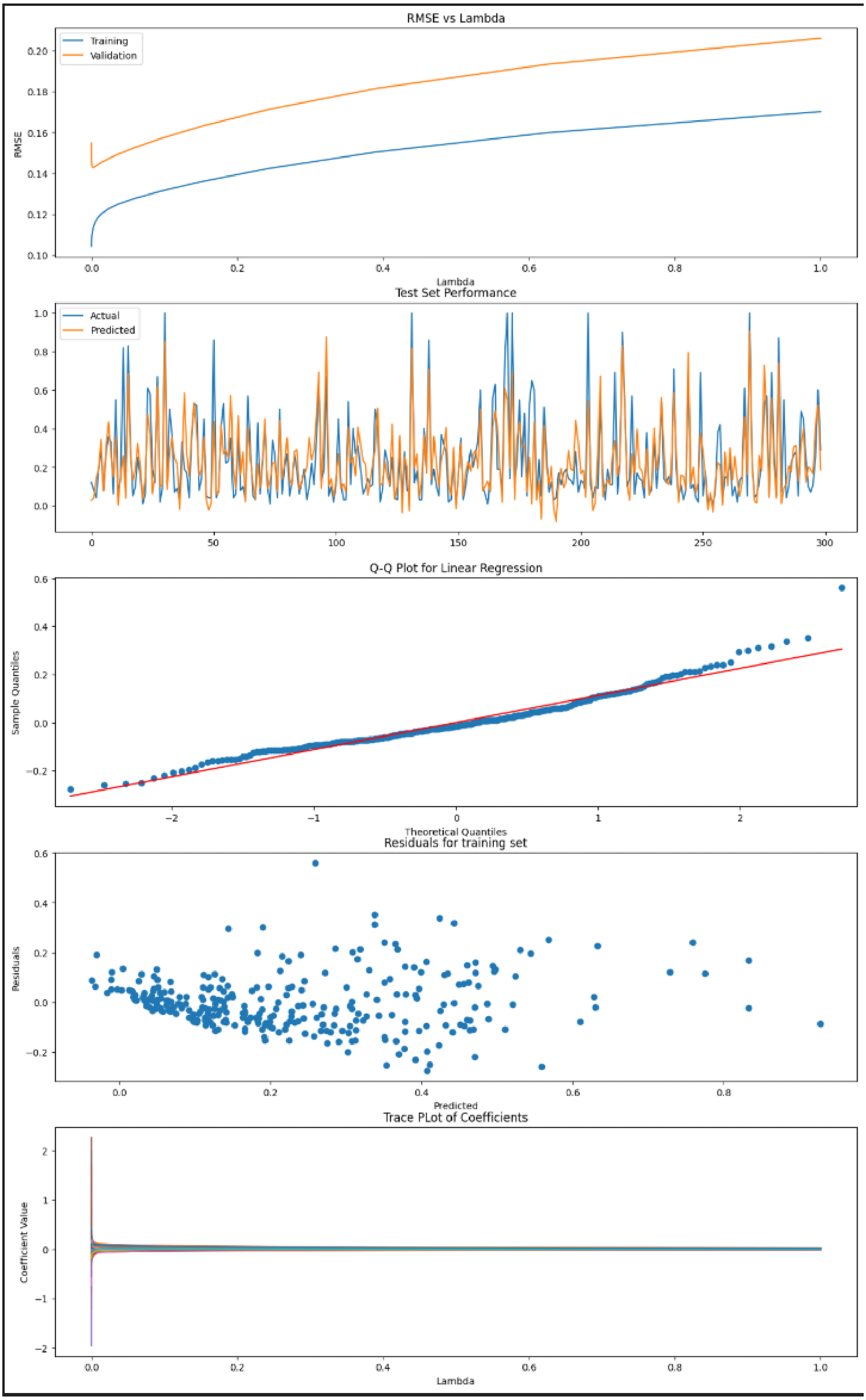


Figure 3: Ridge Model Evaluation

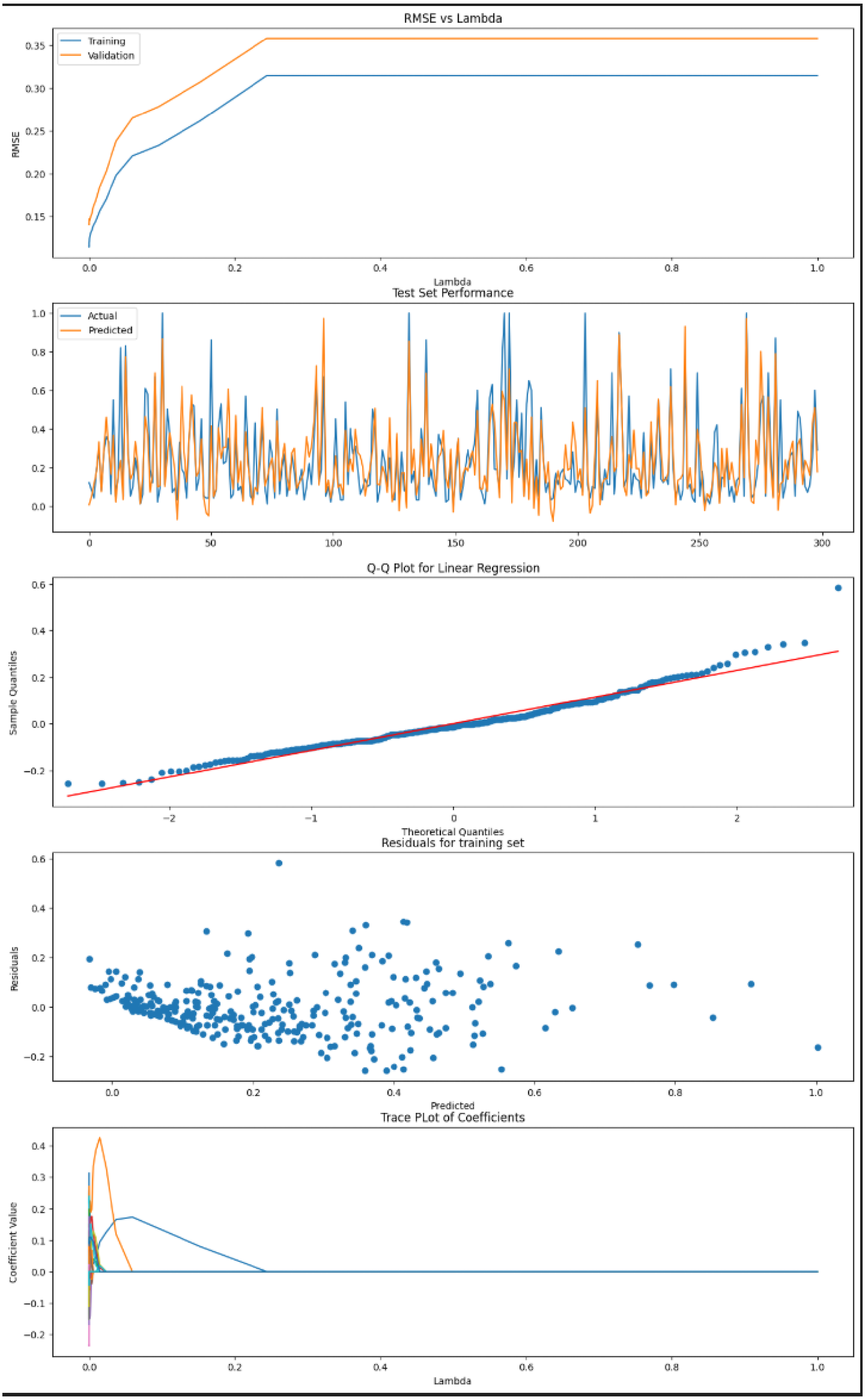


Figure 4: LASSO Regression Evaluation

Conclusion

When comparing the three models, Ridge regression demonstrated the best predicative performance, with the highest test R^2 and the most stable coefficient estimates. Despite LASSO being slightly less accurate, it showed better interpretability through reducing the number of features which makes it ideal for applications where understanding key predictors is important. OLS could only really be useful as a baseline with this dataset, as it suffered from overfitting and lacked the robustness necessary.

Ethical Considerations

There are many ethical considerations to be considered when exploring the link between socio economic factors and violent crimes per capita. Most apparent of which is that the dataset is entirely just collated statistics and does not take into consideration intricate dynamic socio-economic systems that may be present in communities that could potentially introduce biases into the data.

Additionally, biases can be introduced from data encoding choices, which can emphasise a misrepresentation of certain minority groups. This can lead to a distortion of understanding of crime, which in turn reinforces harmful stereotypes of said minority groups. Further, a dataset such as this over-simplifies the relationship between socio-economic factors and crime, overlooking critical factors such as institutional racism, historical trauma, and mental health issues within communities. Without accounting for such complexities, there is a danger of coming to misguided conclusions that focus on surface-level solutions rather than addressing the root causes of crime.

Question 2 – Classification

Data Preprocessing

Standardisation of features through z-score normalisation was the focus of the preprocessing phase of the implementation. The transformation was applied consistently across the training, validation, and test datasets utilising the mean and standard deviation calculated exclusively from the training data. The standardisation of the data can be represented by the following equation:

$$X_{standardised} = \frac{X - \mu}{\sigma}$$

$$where: \mu \text{ } is \text{ } mean, and \text{ } \sigma \text{ } is \text{ } the \text{ } standard \text{ } deviation \text{ } of \text{ } the \text{ } training \text{ } set.$$

This approach to preprocessing was selected for the following reasons:

1. Model requirements – for a distance-based CKNN and margin-based SVM, its critical to ensure equal feature distribution. Despite RF typically being scale irrelevant, standardisation helped with shallow trees.
2. Algorithmic Stability – critical for the proper tuning of the SVM's C value and RBF kernel, and for meaningful CKNN neighbour weighting.
3. Visual Verification – utilising a boxplot visualisation for before and after standardisation showed:
    - All features now centred around 0, with consisted IQR ranges
    - No extreme outliers that could distort kNN distances
    - Equal scaling as needed for proper RF feature importance comparisons

Through calculating the normalisation parameters from only the training set prevents data leakage, which helps to ensure the validation and test sets remain independent evaluations of the model's performance. Additionally, this approach mirrors real-world scenarios where test data statistics are not available during training.

More aggressive methods of preprocessing such as dimensionality reduction were intentionally avoided, as original features contained meaningful discriminatory power.

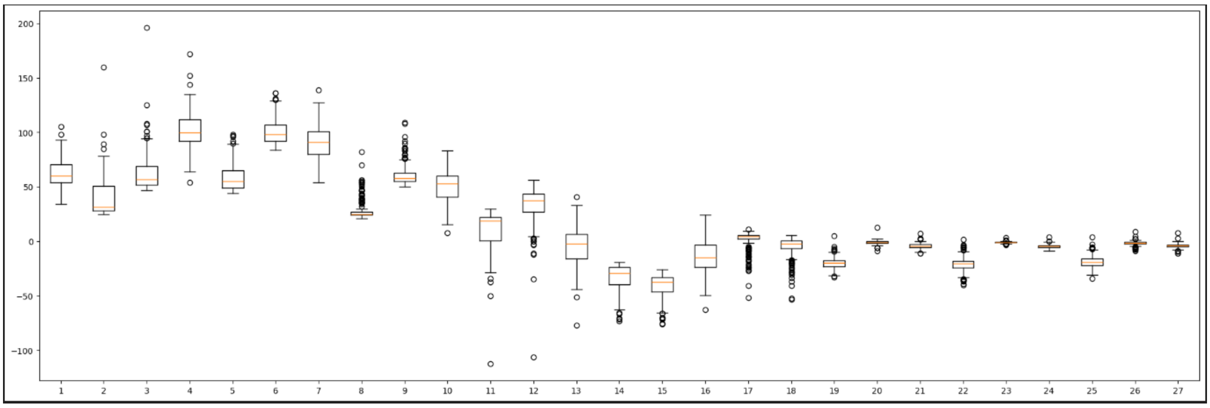


Figure 5: Training Data Before Standardisation

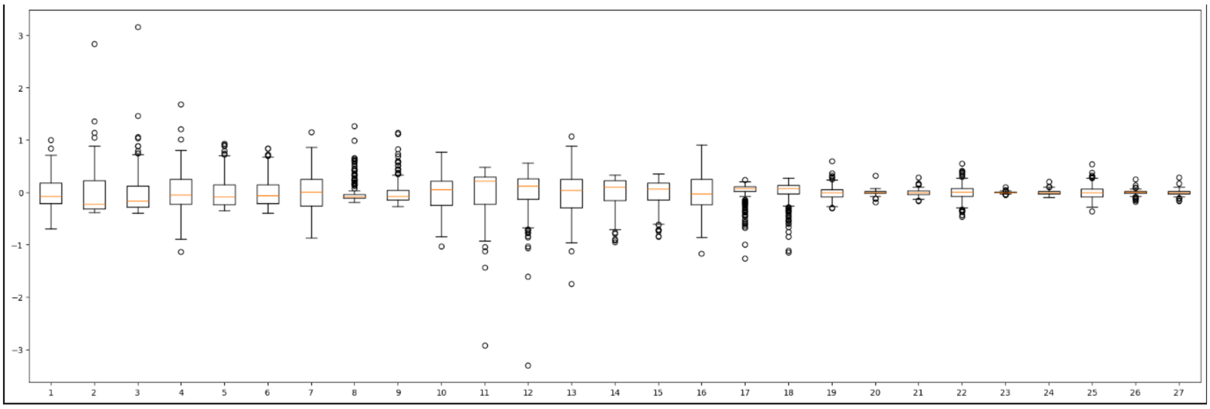


Figure 6: Training Data After Standardisation

Hyperparameter Selection and Model Parameters

CKNN (Best Validation: 87.2%)

The CKNN model achieved a best validation accuracy of 87.2% with an unexpected k-value of 1, using Euclidean distance and uniform weights. This configuration suggests highly localised decision boundaries that potentially prioritise immediate neighbours over broader patterns, which may increase sensitivity to noise or date outliers. While effective on the validations set, the choice of k value being 1 raises concerns of the model overfitting, as the model essentially matches the training set's most proximate examples without generalising to broader trends.

SVM (Best Validation: 85.4%)

For the SVM, its best performance of 85.4% validation accuracy was achieved with a linear kernel and moderate regularisation (c=10) under a one versus one (OVO) decision scheme. The model's selection of linear kernel is indicative that the classification boundaries between target classes are nearly separable through hyperplanes in the feature space, while the average C-value balances maximising the margin and accommodating minor misclassifications. The chosen configuration indicates the underlying data possesses inherent linear characteristics that the SVM model can effectively capture without requiring any complex kernel transformations.

Random Forest (Best Validation: 83.5%)

Random Forest (RF) yielded a best validation accuracy of 83.5% using 100 extremely shallow trees as indicated by the max depth value being 1. While this architecture increases interpretability through generating simple decision stumps. It consequently limits the model's capacity to capture more nuanced and sophisticated feature interactions. The choice of depth being one implies that individual features contain substantial predictive power by themselves, as the model still achieves reasonable performance without needing to build deep hierarchical decision rules. However, the simplicity of the model may result in underfitting if more complex patterns were present in the data.

Data Characteristics Relationship

- **Low k in CKNN**: Indicates very localized class boundaries in feature space
- **Linear SVM Success**: Suggests near-linear separability between classes
- **Shallow RF Trees**: Implies individual features carry strong predictive power

Model Evaluation and Comparison

The SVM demonstrated the best classification capability with a 90% test accuracy followed by CKNN with 86% and then Random Forest with 82%.

Model	Accuracy	Weighted F1	Optimal Parameters	Inference Speed
CKNN	86%	0.86	K=1, Euclidean	Fastest
SVM	90%	0.9	C=10, Linear Kernel	Moderate
Random Forest	82%	0.81	Depth=1, 100 estimators	Slowest

SVM

1. Algorithm Advantages:
    - Linear kernel effectiveness indicates that features are mostly linearly separable as well as negating the need for complex transformations.
    - OVO strategy successfully handles the 4-class imbalance and uneven class distributions.
2. Standardisation Impact:
    - Prevents feature scaling issues
    - Enables proper margin optimisation
    - Supports reliable C parameter tuning
3. Class-Specific performance:
    - Exceptional precision/recall for mixed deciduous (88% correctly classified), Hinoki (91% correctly classified), and Sugi (93% correctly classified).
    - Other class shows 73% accuracy, with the remaining 27% misclassified as mixed deciduous

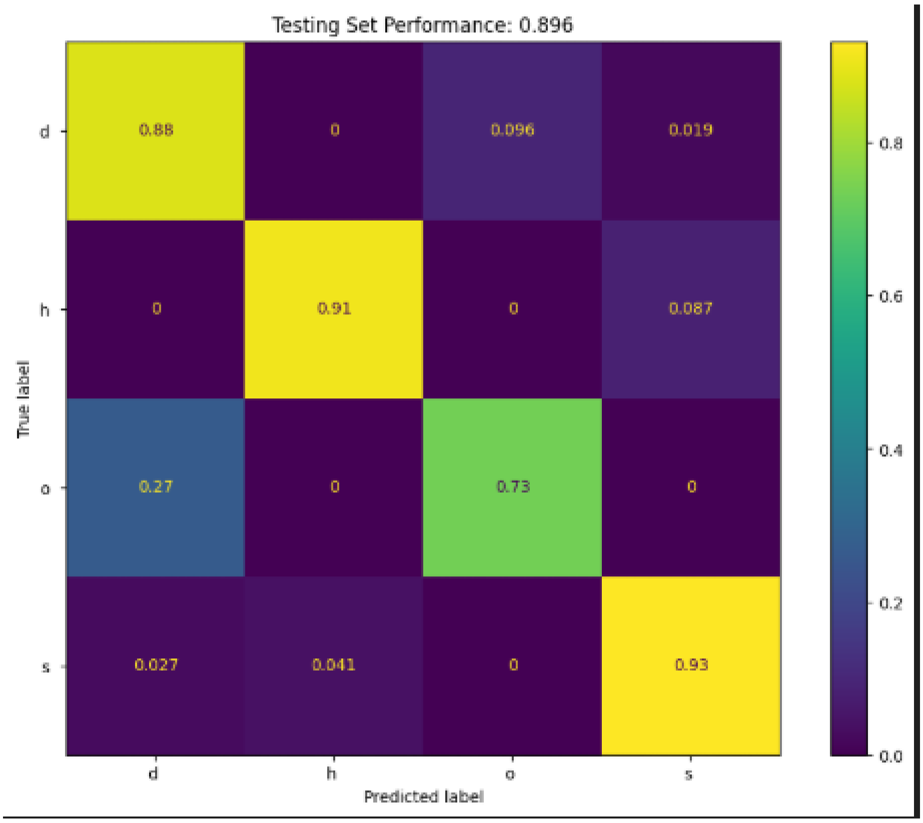


Figure 7: SVM Test Set Confusion Matrix

CKNN

1. Key Strengths:
   - The Sugi forest class had the highest precision with 96%
   - The Hinoki forest class achieved 91% recall despite being the smallest class
   - Strong diagonal dominance in the confusion matrix indicates effective separation
2. Primary Challenges:
   - Significant confusion between ‘other’ and ‘mixed deciduous’ classes (23% of other class samples)
   - Asymmetric error pattern – (23% other to mixed deciduous, 6% mixed deciduous to other)
   - Suggests directional similarity in feature space
3. Parameter insights
   - K=1 parameter selection suggests very localised decision boundaries as well as potential overfitting to training patterns
   - Euclidean distance allows for the effective capture of distinct shapes, but struggles with similar-shaped classes such as other and mixed deciduous

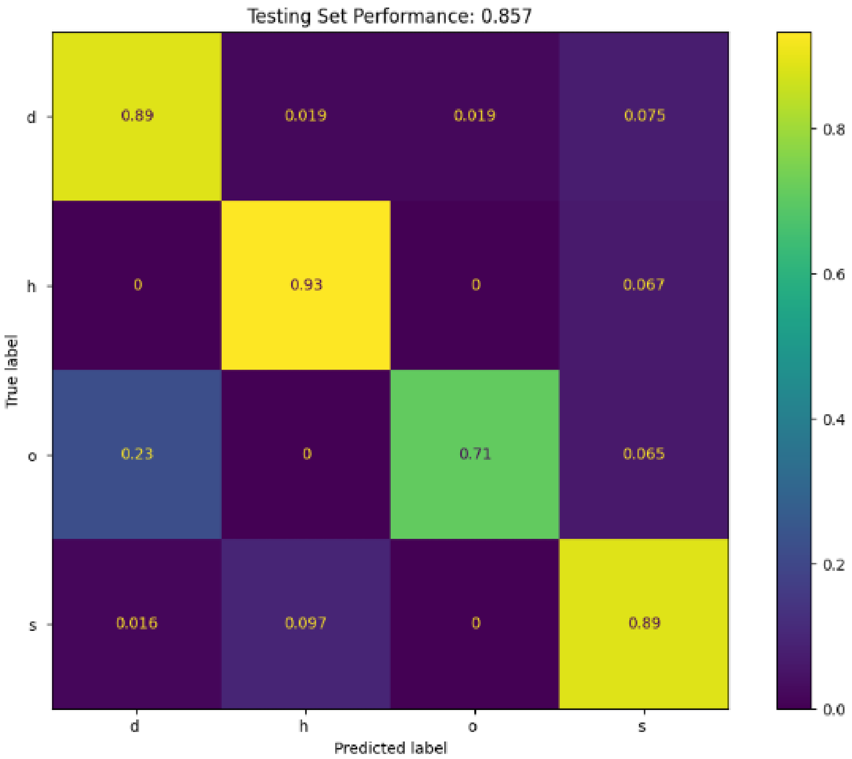


Figure 8: CKNN Test Set Confusion Matrix

Random Forest

2. Top Performers
   - Class 's' dominates with 95% correct classifications
   - Class 'd' shows reasonable performance (71% accuracy)
3. Key Challenges
   - Severe ‘h’ misclassification, with 26% predicted as ‘d’
   - ‘o’ shows problematic confusion with 27% being misclassified as ‘d’, and 13% as ‘s’
4. Depth Limitation impacts
   - Underfitting evident in:
     - High cross-class errors
     - Poor capture of complex features

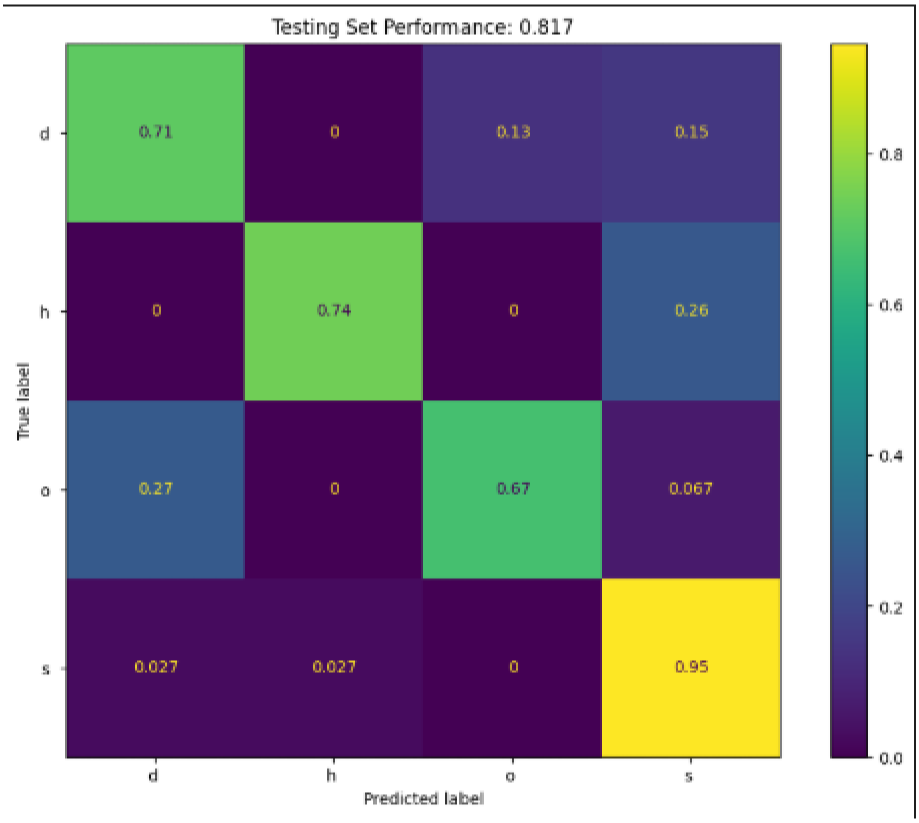


Figure 9: Random Forest Test Set Confusion Matrix

Question 3 – Training and Adapting Deep Networks

Neural Network Design

The chosen neural network was a VGG-style designed with three main considerations in mind: effectiveness for digit recognition, computational efficiency given the small dataset, and regularisation needs.

The neural network was designed with three primary convolutional blocks, with each consisting of two convolutional layers using 3x3 filters with ReLU activations, which was then followed by batch normalisation and max pooling. Dropout is applied progressively from 0.2 to 0.4 in 0.1 increments across the blocks as a mean to mitigate overfitting. The classifier head includes a flattening layer, which is a 256-unit dense layer with ReLU activation and 0.5 dropout, followed by a 10-unit softmax output layer for multi-class classification. With this architecture, it allows for a good balance of computational efficiency and feature extraction capability and the utilisation of small 3x3 filters to capture local patterns while batch normalisation stabilises training. The moderate depth of three convolutional blocks was chosen to avoid excessive complexity, making the model more suitable to be run on computers with limited computational resources.

The Adam optimiser was utilised in the training process with a learning rate of 1e-3 and categorical cross-entropy loss which was monitored through accuracy, precision, and recall metrics. The baseline model used a batch size of 64 and trained up to a maximum of 100 epochs. This approach achieved a 82.47% test accuracy in approximately 230 seconds. With the augmented model, it was trained with a smaller batch size of 32 but was extended to use up to 150 epochs. In addition to this, the model incorporated random rotations, zooms, translations, contrast adjustments, and gaussian noise. While the augmentation improved test accuracy to 83.69% and precision to 91.7%, recall was reduced by 3.6%, which suggests the model became more conservative in its predictions. Training time also increased to 357 seconds due to the computational overhead of on-the-fly augmentation.

The models were constrained by single-GPU training, which limits the feasibility of deeper architectures such as ResNet or large batch sizes. Despite this limitation, the implemented design achieved strong performance, with augmentation particularly benefitting challenging classes such as 8. However the trade-off between precision and recall indicates potential for further refinement such as class-weighted loss to address class imbalance and hard examples. Overall, the network demonstrates that a carefully regularised CNN can achieve good results with the SVHN dataset within practical hardware limits.

Data augmentation

The given data was augmented to address the challenges of Street View House Numbers (SVHN) while preserving label integrity. Four key transformations were implemented: rotation (±10%), translation (±10%), zoom (±20%), and gaussian noise (σ=0.01). these parameters were specifically tuned via iterative testing to ensure they generation realistic variations without introducing excessive distortion. Vertical and horizontal flipping of images was strictly forgone, due to some digits having altered meanings when doing so (e.g. 6 and 9). The selected augmentations allow for an effective simulation of natural geometric and photometric variations present in street view images while respecting the properties of digit classification, which provides valuable regularisation for the limited training set. With this approach, training distribution was successfully expanded without introducing any synthetic artifacts that could mislead the network.

Comparison of DCNNs

Performance Metrics

The models substantially outperformed the SVM across all classification metrics. The augmented VGG model achieved the highest test accuracy of 83.7%, precision of 91.7% and F1-score of 0.84. closely behind was the baseline VGG with 82.7% accuracy and an F1 score of 0.82. in contrast, the SVM achieved only 19% accuracy, and an average F1 score of 0.14 which clearly highlights its poor suitability for complex image recognition tasks.

Despite both VGG models demonstrating balanced class-wise performance, the augmented model showed greater generalisation, particularly in underrepresented or difficult classes such as 8 and 9. This improvement was evident in the learning curves and confusion matrices, where the augmented model exhibits higher validation accuracy and reduced overfitting compared to the baseline model.

Training and Inference Time

The SVM trained significantly faster (1.25 seconds) than both DCNNs, though the incredibly poor performance makes the SVM less attractive even in rapid prototyping or low-resource environments. The baselines VGG model took approximately 230 seconds to train, whereas the augmented version took about 356 seconds, which reflects the additional computational cost of augmenting the data. The inference time was essentially the same for both models, with the baseline getting 1s 3ms/step and the augmented getting 1s 4ms/step.

Evaluation Summary

Despite the higher computational needs of Deep Convolutional Neural Networks, they provide dramatically better performance than the SVM. The improvements to generalisation and class-wise consistency effectively demonstrate the value of data augmentation, which helps to increase model robustness and mitigate overfitting. Despite the SVM offering speed advantages, the lack of representational power makes it highly unsuitable for large-scale image classification in deep learning contexts.

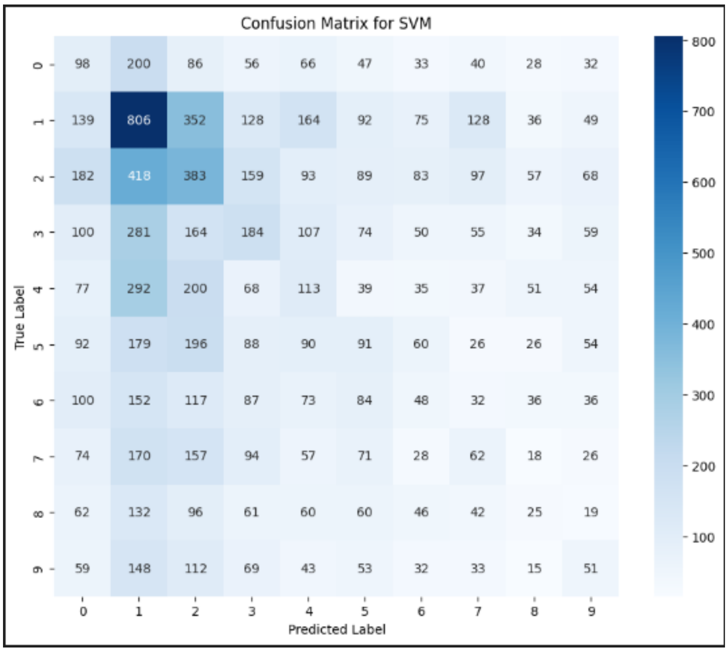


Figure 10: Confusion Matrix for SVM

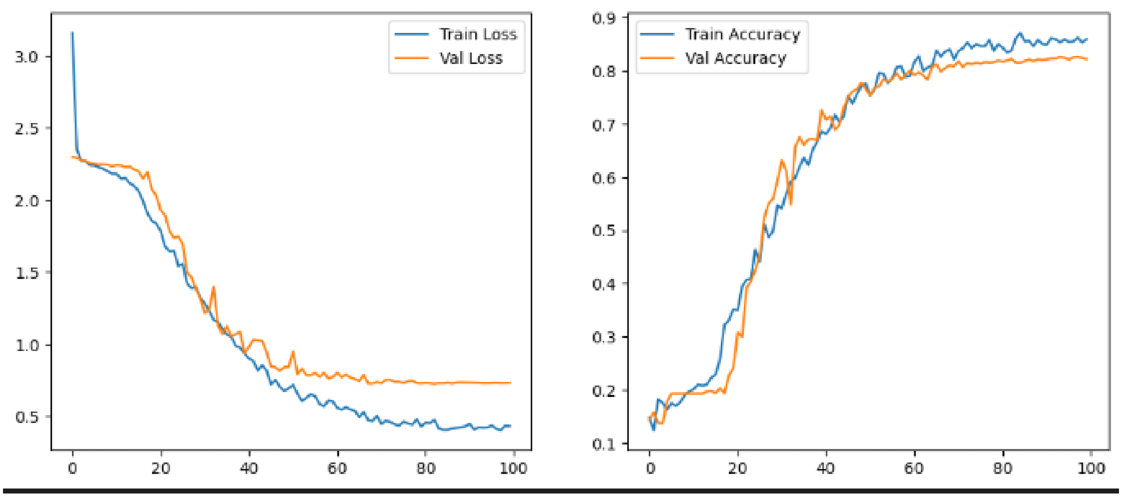
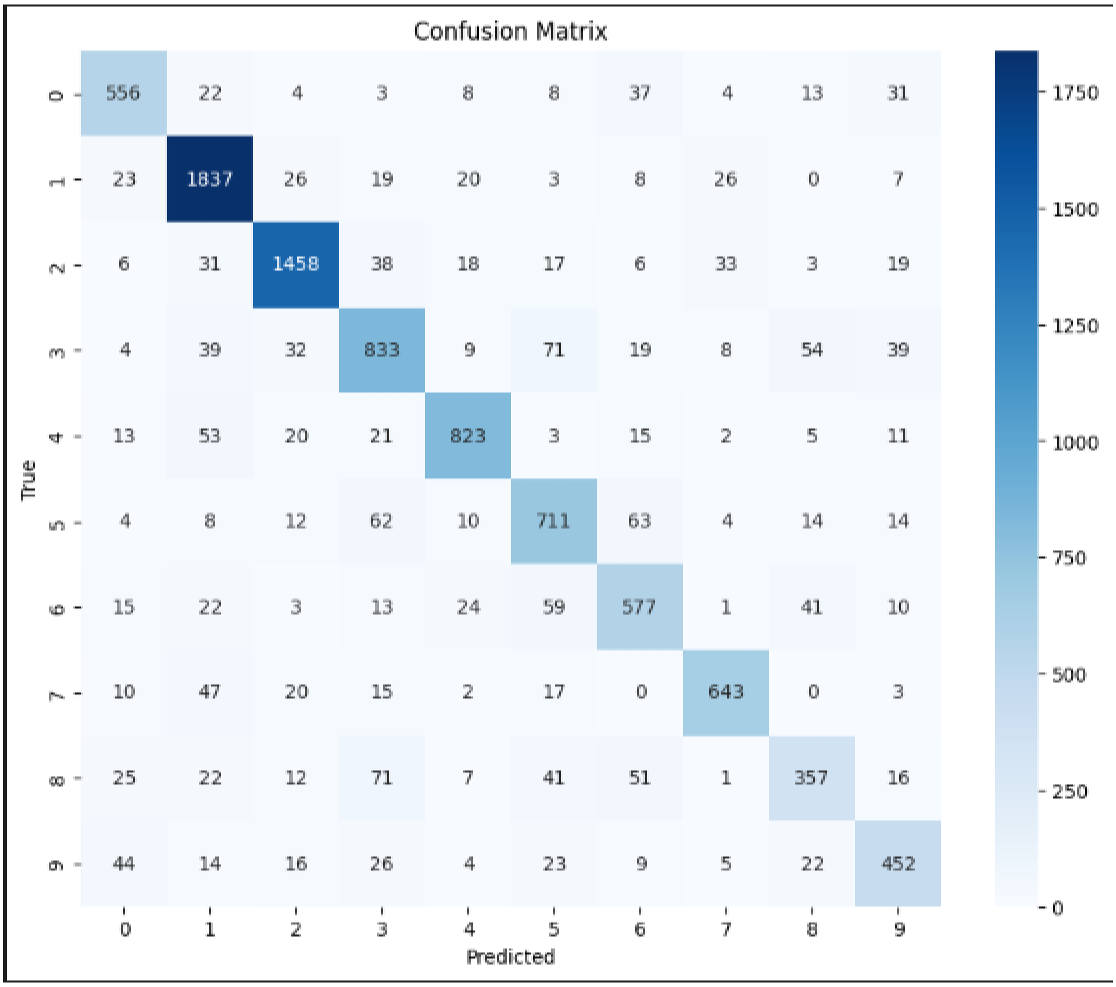


Figure 11: Baseline VGG Train and Validation Loss, Train and Validation Accuracy



True \ Predicted	0	1	2	3	4	5	6	7	8	9
0	556	22	4	3	8	8	37	4	13	31
1	23	1837	26	19	20	3	8	26	0	7
2	6	31	1458	38	18	17	6	33	3	19
3	4	39	32	833	9	71	19	8	54	39
4	13	53	20	21	823	3	15	2	5	11
5	4	8	12	62	10	711	63	4	14	14
6	15	22	3	13	24	59	577	1	41	10
7	10	47	20	15	2	17	0	643	0	3
8	25	22	12	71	7	41	51	1	357	16
9	44	14	16	26	4	23	9	5	22	452

Figure 12: Confusion Matrix for Baseline VGG

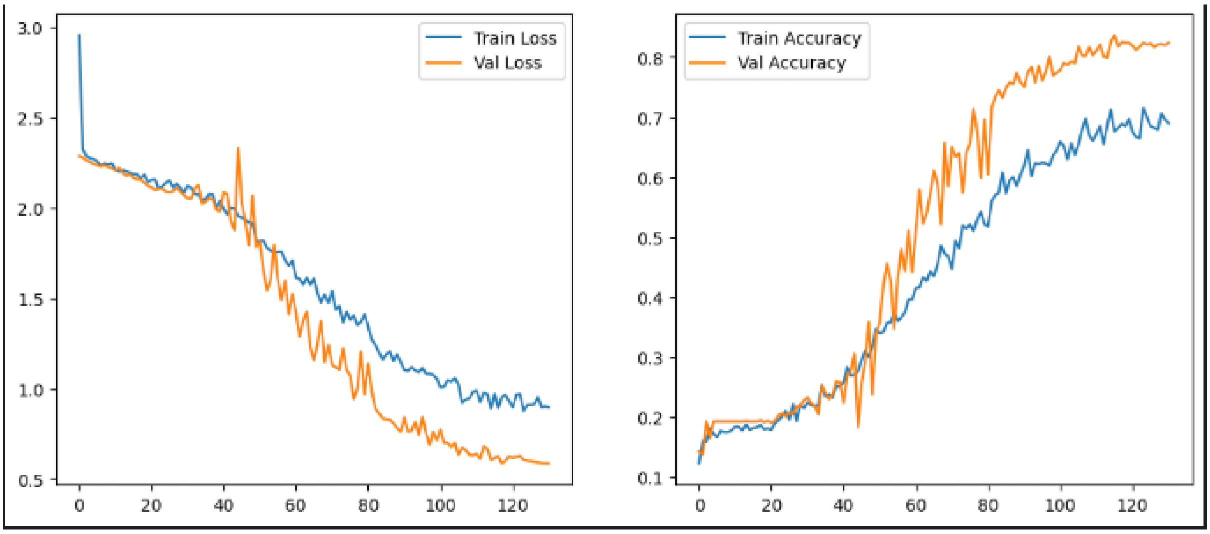
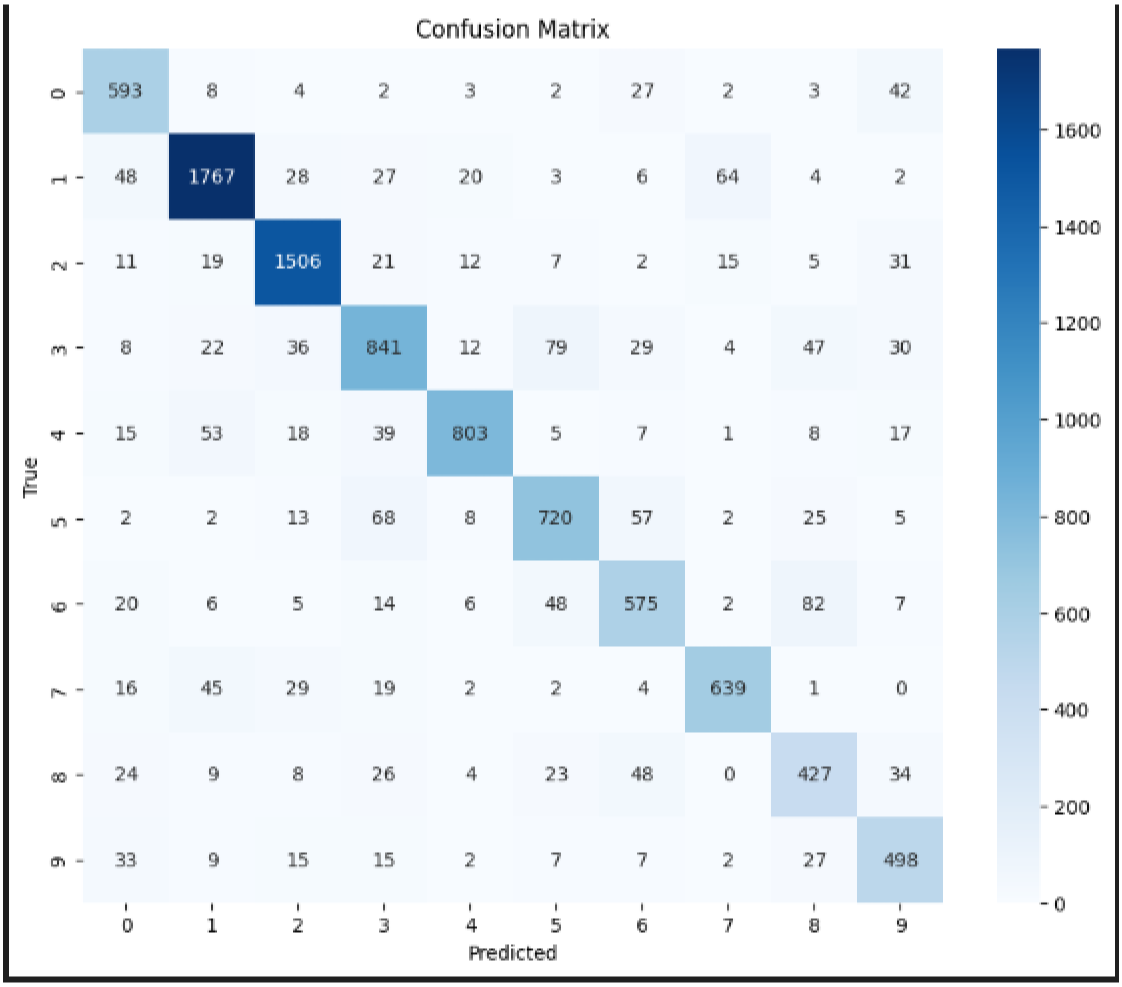


Figure 13: Augmented VGG Training and Validation Loss, Training and Validation Accuracy



	0	1	2	3	4	5	6	7	8	9
0	593	8	4	2	3	2	27	2	3	42
1	48	1767	28	27	20	3	6	64	4	2
2	11	19	1506	21	12	7	2	15	5	31
3	8	22	36	841	12	79	29	4	47	30
4	15	53	18	39	803	5	7	1	8	17
5	2	2	13	68	8	720	57	2	25	5
6	20	6	5	14	6	48	575	2	82	7
7	16	45	29	19	2	2	4	639	1	0
8	24	9	8	26	4	23	48	0	427	34
9	33	9	15	15	2	7	7	2	27	498

Figure 14: Confusion Matrix for Augmented VGG