Abdullah Sharaf (2139533)

1. Introduction

Sorting pictures is an important job in computers that can be used for recognizing things or checking medical images. In this study, we look at how to do this using two different methods: Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs). Our main goal is to see how well they work on a smaller part of the CIFAR-10 dataset. The main problem is giving specific labels to pictures based on what's in them. This is important for things like self-driving cars, healthcare, and security. The hard part is making models that can sort pictures correctly across many different groups. Our idea to solve this problem is to compare CNNs, which are good at learning features, with SVMs using Histogram of Oriented Gradients (HOG) features. We want to understand what each method is good at and not so good at and figure out when to use them. We plan to test both models on 5,000 training and 1,000 test pictures from the CIFAR-10 dataset. This dataset has lots of different pictures. We'll look at the results to see how well each method works, using things like accuracy, confusion charts, and classification reports.

2. Method

2.1 CNN Model

The CNN model is a sequential architecture comprising convolutional layers tech patterns and images and detect pattern well. And it need number of filters ,activation functions, pooling layers, dropout layers, and dense layers. We will iterate 12 times Trained on the CIFAR-10 subset, the CNN automatically learns features during training. The Adam optimizer is employed with a learning rate of 0.001. The model is evaluated using categorical cross-entropy loss. In the CNN approach, the architecture is designed to capture hierarchical features through convolutional layers.

The specifics of the architecture are as follows:

Convolutional Layers: Two sets of convolutional layers with 32 and 64 filters, respectively, each having a kernel size of (3, 3). The use of convolutional layers allows the model to learn spatial hierarchies in the image. Activation Functions: ReLU (Rectified Linear Unit) activation functions are applied after each convolutional layer. ReLU introduces non-linearity to the model, enabling it to learn complex patterns. MaxPooling Layers: MaxPooling layers with a pool size of (2, 2) are added to reduce the spatial dimensions, providing a form of down-sampling. Dropout Layers: Dropout layers with a dropout rate of 0.25 are inserted to prevent overfitting during training. Dropout randomly drops a certain percentage of neurons during each training iteration. Flatten Layer: After the convolutional layers, a flatten layer is introduced to transform the 2D output into a vector, preparing it for the fully connected layers. Dense Layers: Two dense layers are added with 512 neurons each, followed by ReLU activation functions. The final dense layer has 10 neurons corresponding to the 10 classes in CIFAR-10, with a softmax activation function for multi-class classification. Training and Optimization: The model is trained using the Adam optimizer with a learning rate of 0.001. The categorical cross-entropy loss is employed as the loss function.

2.2 SVM model

The SVM model utilizes HOG features extracted from the CIFAR-10 images. The preprocessing involves converting RGB images to grayscale, normalizing pixel values, and computing HOG features. A pipeline with standard scaling, PCA for dimensionality reduction, and an RBF kernel SVM with hyperparameter C=10 is employed. Training and evaluation are performed on the same CIFAR-10 subset. The SVM approach involves feature extraction using a Histogram of Oriented Gradients (HOG) and subsequent training with a Support Vector Machine.

The detailed steps are as follows:

Abdullah Sharaf (2139533)

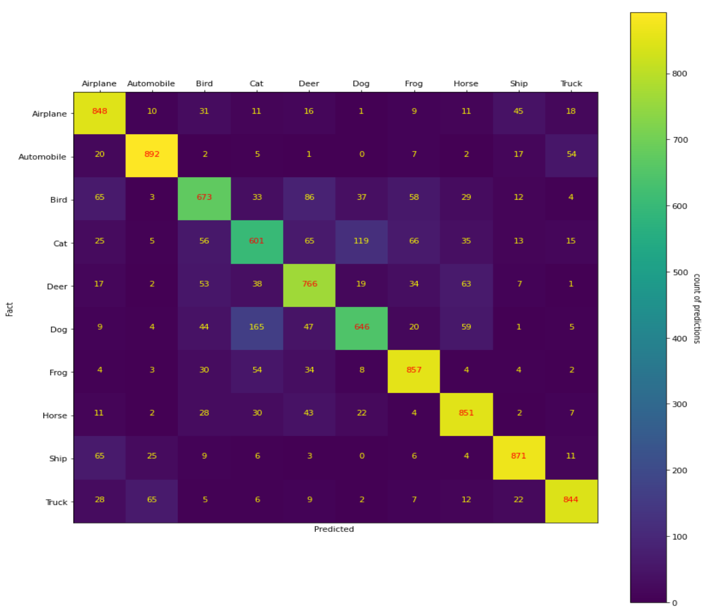
Image Preprocessing: RGB images are converted to grayscale to reduce the dimensionality and computational complexity. Pixel values are normalized to the range [0, 1]. HOG Feature Extraction: HOG features are extracted from each grayscale image. This involves computing gradients in pixel intensity and grouping gradient information in local regions. The resulting feature vector captures the distribution of gradients in different parts of the image. Standard Scaling: The feature vectors obtained from HOG are standardized using standard scaling to ensure that all features have a mean of 0 and a standard deviation of 1. Principal Component Analysis (PCA): PCA is employed for dimensionality reduction. This step is crucial for managing the curse of dimensionality and improving the SVM's efficiency. SVM Training: The preprocessed and reduced feature vectors are used to train an SVM with an RBF kernel. The hyperparameter C is set to 10, influencing the trade-off between decision boundary complexity and classification accuracy. Evaluation: The trained SVM is evaluated on the same CIFAR-10 subset used for training, and performance metrics such as accuracy, confusion matrix, and classification report are generated.

3. Results

the detailed methodology provides insights into the intricacies of implementing CNNs and SVMs for image classification. While CNNs exhibit superior accuracy by automatically learning hierarchical features, SVMs with HOG features offer an interpretable and computationally efficient alternative. The choice between the two depends on factors such as computational resources, interpretability, and dataset characteristics.

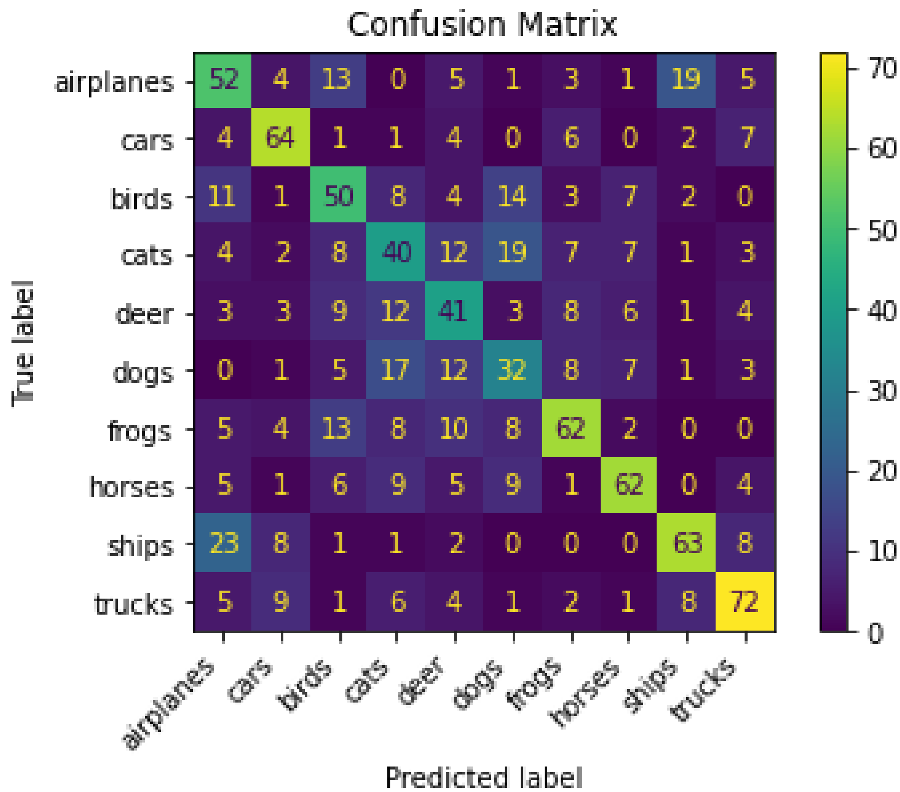
3.1 CNN Results

The CNN achieves a test accuracy of 55.4% on the CIFAR-10 subset. The confusion matrix and classification report provide a detailed breakdown of performance across different classes. Precision, recall, and F1-score metrics offer insights into the model's strengths and weaknesses.



3.2 SVM Results

The SVM achieves a test accuracy of 53.8%. The confusion matrix and classification report illustrate the model's performance on individual classes. Visualization of 20 classified images enhances qualitative understanding.



Future work should explore hybrid models combining CNNs and SVMs for enhanced performance. Additionally, further investigations into hyperparameter tuning, transfer learning for CNNs, and handling class imbalance are recommended. Cross-dataset evaluations on larger and diverse datasets would provide a more comprehensive understanding of model generalization. This study contributes valuable insights into the strengths and limitations of CNNs and SVMs for image classification, guiding practitioners in selecting appropriate models for specific applications. Continued research in this domain is essential for advancing the field of computer vision and improving the applicability of image classification techniques.

4. Conclusion

In conclusion, both CNNs and SVMs are effective for image classification, with CNNs outperforming SVMs in this specific scenario. The CNN's automatic feature learning capability contributes to its higher accuracy. However, the computational complexity of CNNs should be considered, and further experimentation with model architectures and transfer learning could improve results. The computational complexity of CNNs poses a challenge, requiring significant resources. SVMs, while computationally efficient, may struggle with capturing complex image patterns. Model hyperparameters and architecture choices significantly impact performance, necessitating systematic tuning. The field of image classification is dynamic, and future research can explore emerging techniques such as attention mechanisms, neural architecture search, and advancements in transfer learning. Additionally, investigating interpretability tools for CNNs and exploring novel feature extraction methods could further enhance the capabilities of image classification models. This comprehensive methodology provides a foundation for understanding, implementing, and extending CNNs and SVMs in image classification, contributing to the continuous evolution of computer vision applications.

Abdullah Sharaf (2139533)

References

1- Alzubaidi, L., Zhang, J., Humaidi, A.J. et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data 8, 53 (2021). https://doi.org/10.1186/s40537-021-00444-8.

2- Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE. A survey of deep neural network architectures and their applications. Neurocomputing. 2017;234:11–26.

3- Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: an overview and application in radiology. Insights Imaging 9, 611–629 (2018). https://doi.org/10.1007/s13244-018-0639-9.

4- Cristianini, N., Ricci, E. (2008). Support Vector Machines. In: Kao, MY. (eds) Encyclopedia of Algorithms. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30162-4_415

5- Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, Cambrigde, Book website: www.support-vector.net (2000)

6- Hastie, T., Rosset, S., Tibshirani, R., Zhu, J.: The entire regularization path for the support vector machine. J. Mach. Learn. Res. 5, 1391–1415 (2004)