Image Classification: CIFAR-10

Mateusz Marcol, Will Parnell

July 14, 2025

Table of Contents

CIFAR-10: Introduction

- 60,000 labelled images
- Train:Test = 5:1
- Image format: 32x32 RGB
- 10 classes & 6,000 images per class

Frog

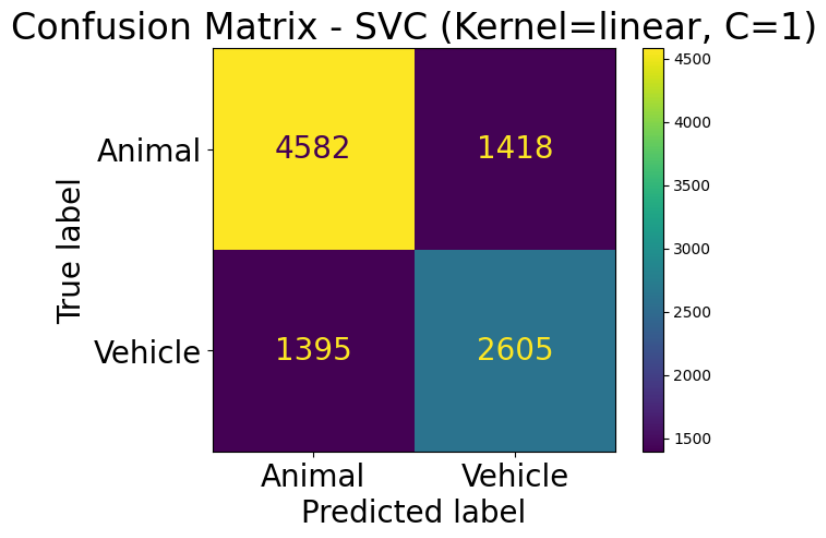
Truck

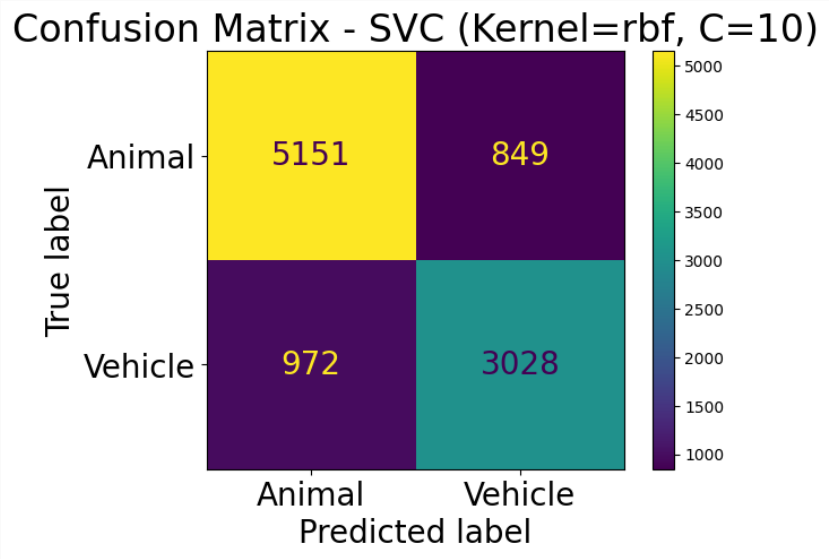
Deer

Method: SVM

- Stratified subset of the training data (1,000 colour images)
- 5-fold cross-validation with grid search on C
- $C \in \{1, 10, 100\}$
- **Kernel**: Linear | RBF
- Measure accuracy on full test set (10,000 images)
- **Tools:** SciKit Learn, NumPy

Results: SVM





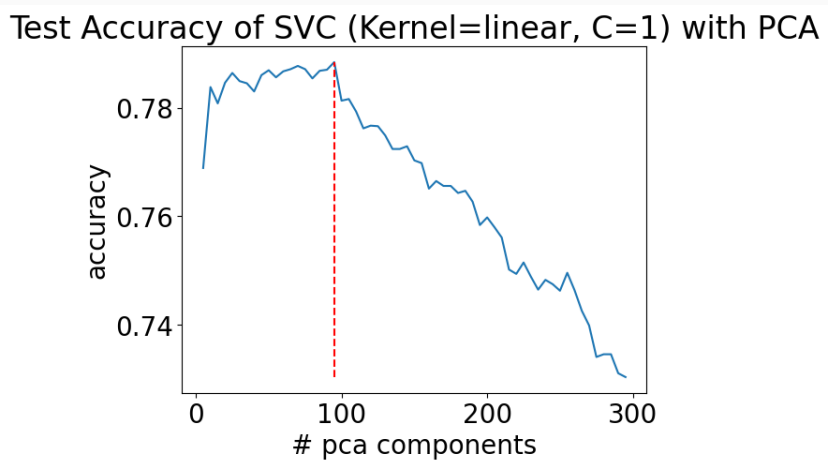
Linear: Accuracy = 71.87%

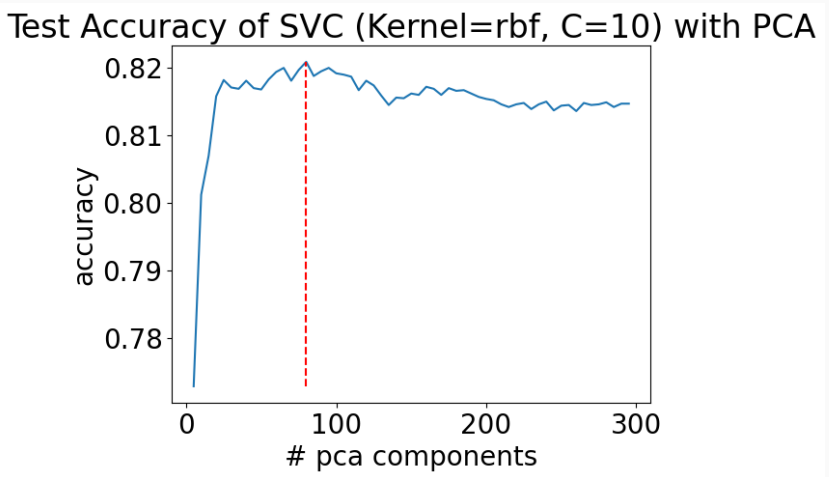
RBF: Accuracy = 81.79%

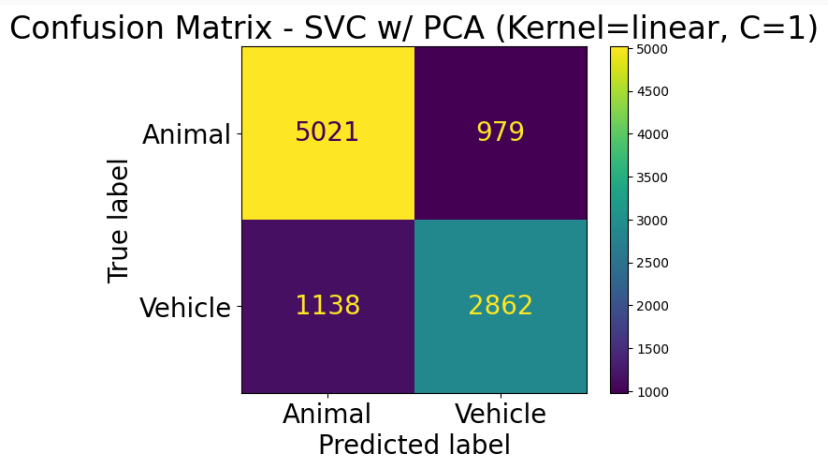
Method: SVM w/ PCA

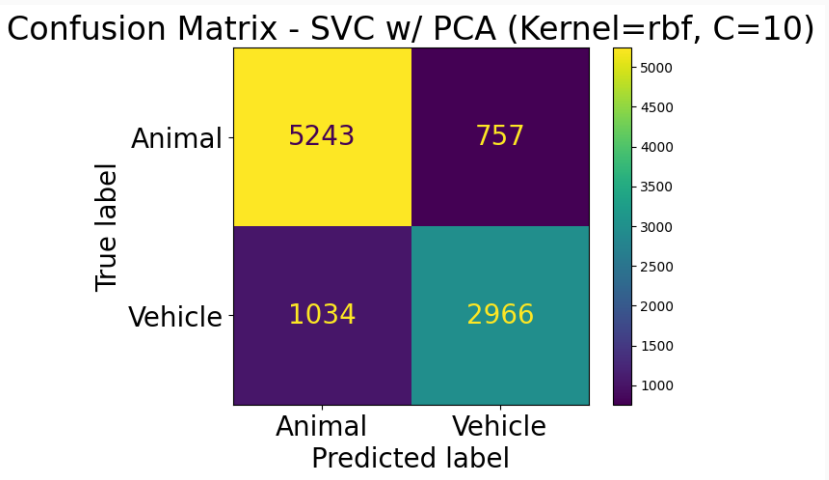
- **Goal:** Find the optimal number of principle components
- Stratified subset of the training data (1,000 images)
- Transform images by PCA
- Set $C = 1 \mid 10$ and Kernel $=$ Linear $\mid$ RBF
- For $p = 5, 10, \ldots, 300$:
  - Take the first p PCs of transformed images
  - Fit SVM classifier to the compressed data
  - Compute test accuracy on full test set
- **Tools:** SciKit Learn, NumPy

Results: SVM w/ PCA (PC Plot)









Linear: $p^* = 95$, Best Accuracy = 78.83%

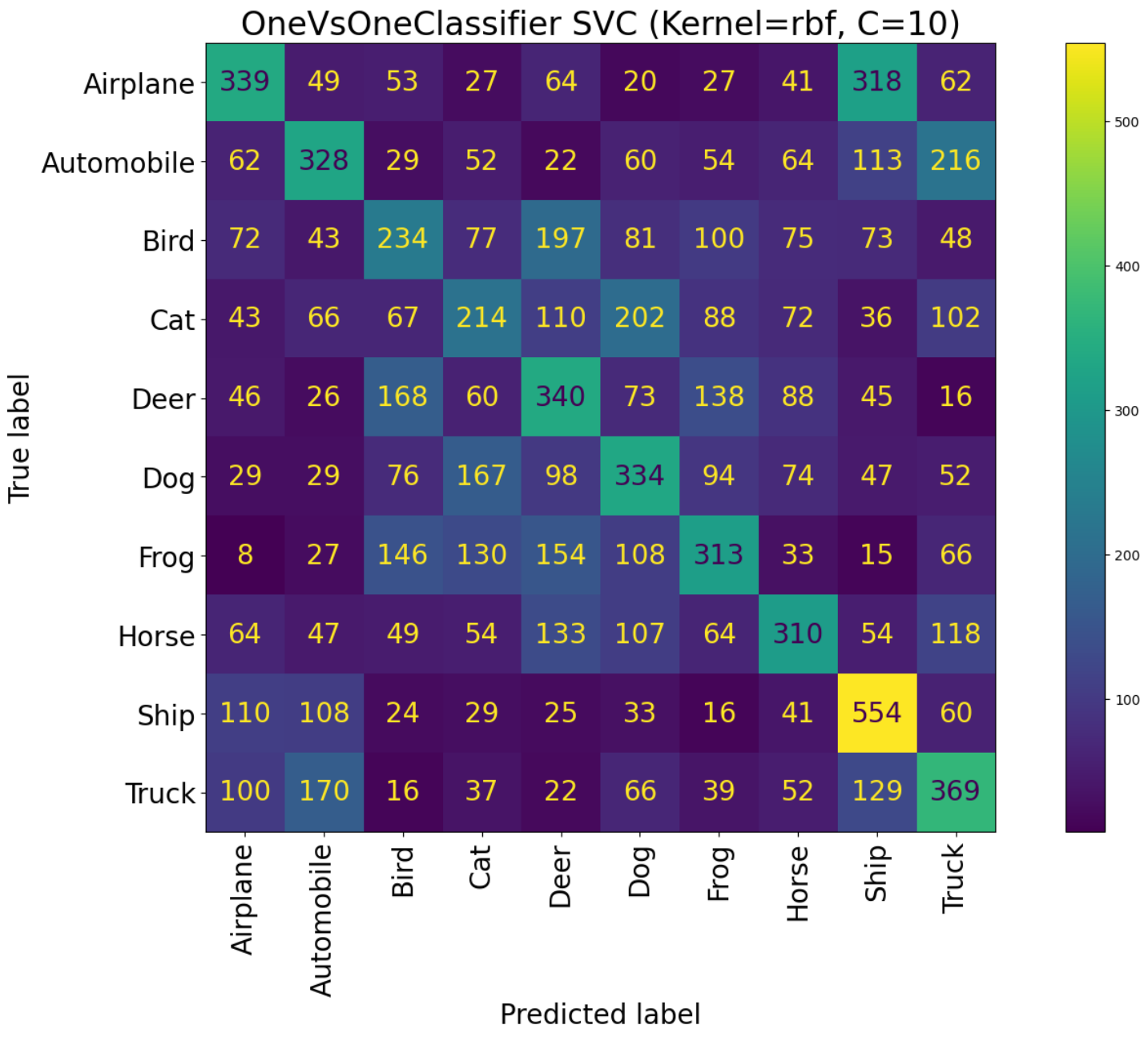
RBF: $p^* = 80$; Best Accuracy = 82.09%

Method and Results: SVM (w/ PCA)

- 1% stratified subset
- C=1 for Linear, C=10 for RBF
- Accuracy:

	Linear	RBF	RBF w/ PCA
OneVsRest	22.74%	32.93%	–
OneVsOne	26.30%	**33.35%**	32.58%

Results: SVM (Best Confusion Matrix)



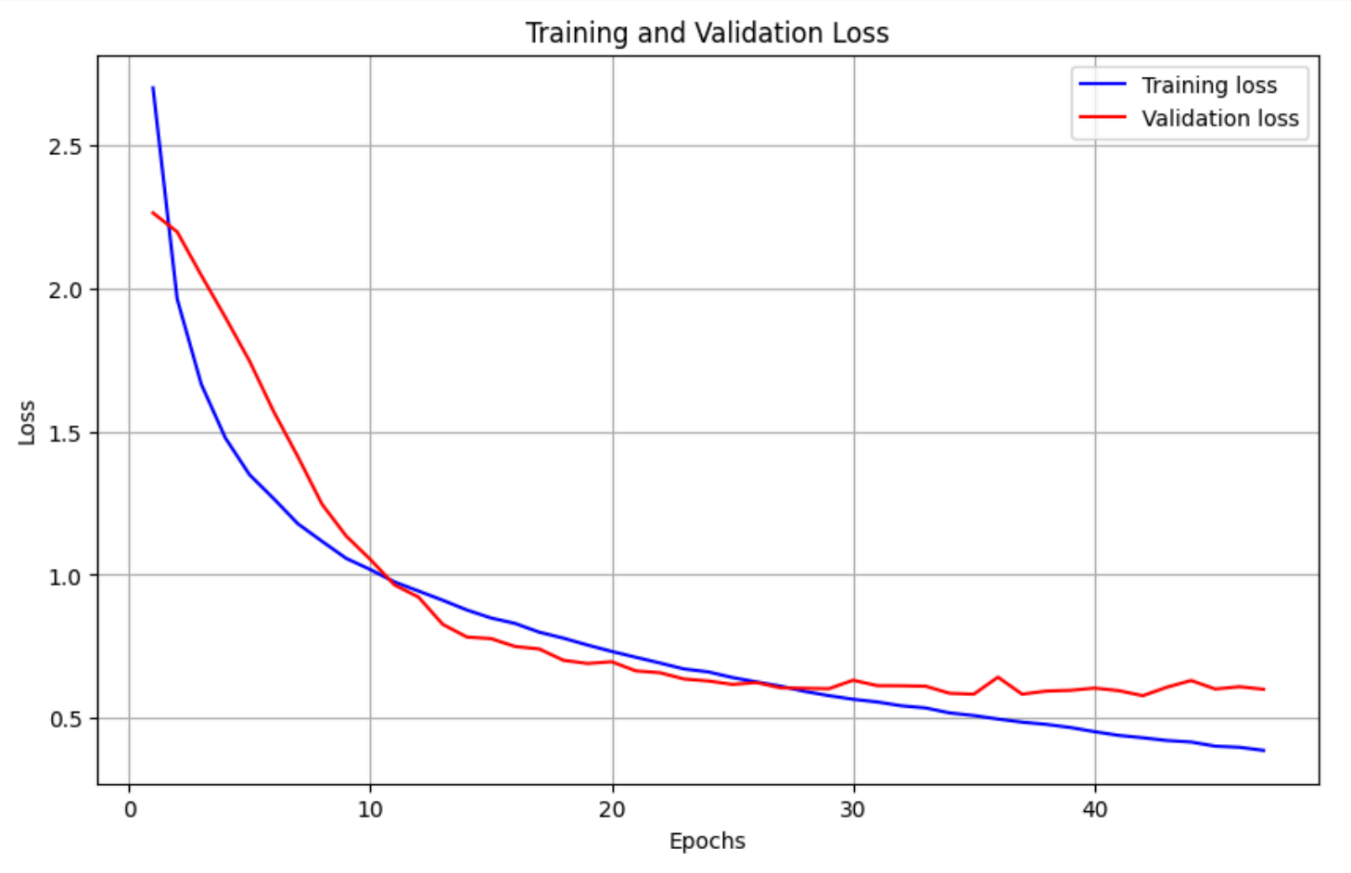
CNN: Architecture

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 32, 32, 24)	672
conv2d_5 (Conv2D)	(None, 32, 32, 40)	8,680
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 40)	0
dropout_3 (Dropout)	(None, 16, 16, 40)	0
conv2d_6 (Conv2D)	(None, 16, 16, 78)	28,158
batch_normalization_1 (BatchNormalization)	(None, 16, 16, 78)	312
conv2d_7 (Conv2D)	(None, 16, 16, 68)	47,804
max_pooling2d_3 (MaxPooling2D)	(None, 8, 8, 68)	0
dropout_4 (Dropout)	(None, 8, 8, 68)	0
flatten_1 (Flatten)	(None, 4352)	0
dense_2 (Dense)	(None, 256)	1,114,368
dropout_5 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 10)	2,570

Total params: 1,202,564 (4.59 MB)
Trainable params: 1,202,408 (4.59 MB)
Non-trainable params: 156 (624.00 B)

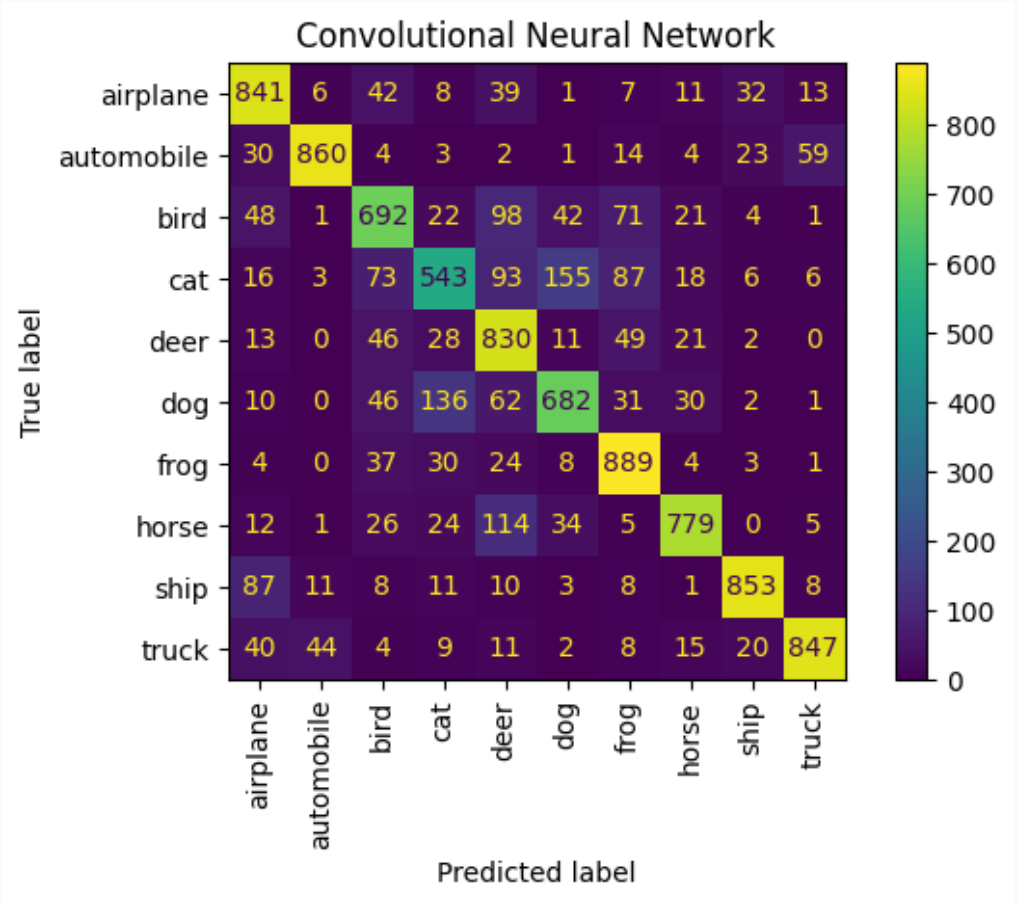
CNNs: Training

- Hyperparameter tuning using `keras_tuner` and `RandomSearch`



CNNs: Results

- **80.19%** accuracy achieved on the test dataset.
- Result is similar to the baseline presented on the authors' website
- CNN is significantly better than SVM



Summary

- SVMs good enough at binary classification of images
- SVMs bad at multi-classification of images
- CNNs excel at image classification in comparison to SVMs
- PCA reduces training of SVMs
- SVMs benefit from PCA preprocessing for binary classification, but not for multi-classification

Any Questions?

Questions?