Deep Learning for Billboard Classification

By

Sayali Avinash Chavan **Dermot Kerr** **Sonya Coleman** **Hussein Khader***

Intelligent Systems Research Centre

University of Ulster, UK

*The Neuron, Amman, Jordan

email:{chavan-s, d.kerr, sa.coleman}@ulster.ac.uk, hussein.khader@theneuron.com

About Me: Sayali

LinkedIn GitHub

PhD Researcher Ulster University, UK

- Computer Vision for Advertising Analytics
- Studentship: Digital Natives Postgraduate Scholarship (International Award)
- Publication: [Billboard Detection in the Wild]

Seminar Support staff

Data Analytics

Business School Ulster University, UK

PG: MSc in Advanced Computer Science

Loughborough University, UK

- British Council Ambassador – Full Scholarship
- Dissertation : Machine Learning for Wildlife Conservation

Associate Fellowship by Advance HE

- Level 7 Postgraduate CPD Framework UK
- Award: First Steps to Supporting Learning and Teaching in Higher Education (FST)

UG: Bachelors of Engineering in Electronics & Telecommunication

- Pune University - Dean list rank.

[SPPU]

Pune University

Application Development Analyst Accenture, India

Introduction

Advertising is essential to increase product awareness and foster a positive outlook, which in turn helps sales.

The purpose of this study is to create a classifier for Billboard images using CNN model that can accurately categories billboards amidst other types of images in natural environments.

The goal is to train 5 CNN architecture using billboard images merged with subset of CIFAR10 dataset, evaluate their effectiveness and determine the most efficient deep learning model.

Dataset

CIFAR10 In total - 60,000 colour images 30*30 pixels : Airplane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship, and Truck. (700/class)

Billboard images (778) were pre-processed to match the CIFAR10 standards, creating two separate datasets:

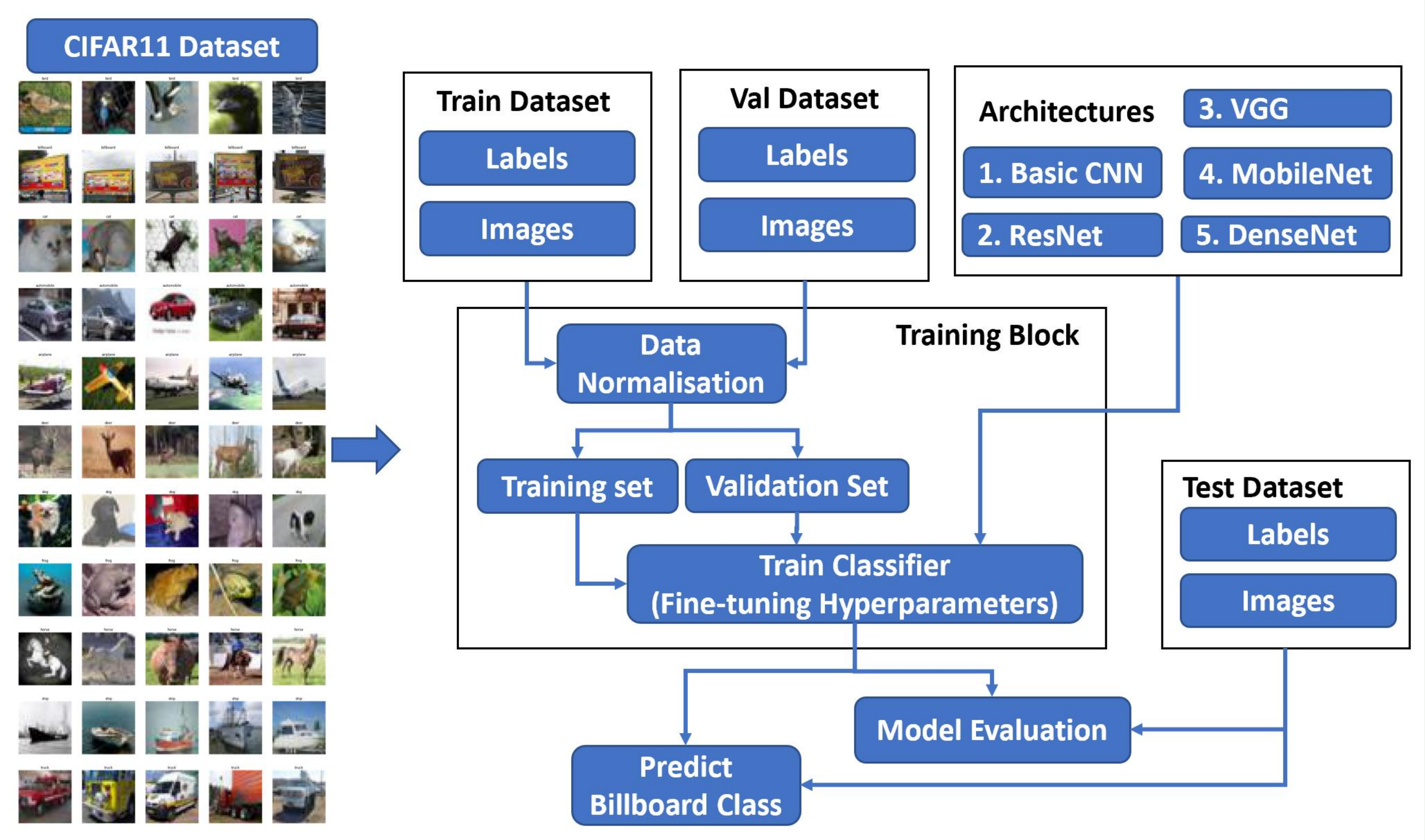
CIFAR11: Billboard class combined with all 10 classes of CIFAR10 - 6188 images used for the training set, 1546 images for validation set and 220 images for the testing set.

Total : 7,954

CIFAR2: Based on the outcomes of CIFAR11, the ’Billboard’ class was merged with only ’Ship’ class from CIFAR10 - 1230 images used for the training set, 306 images for the validation set and 40 images for the testing set.

Total: 1,576

Billboard Classifier

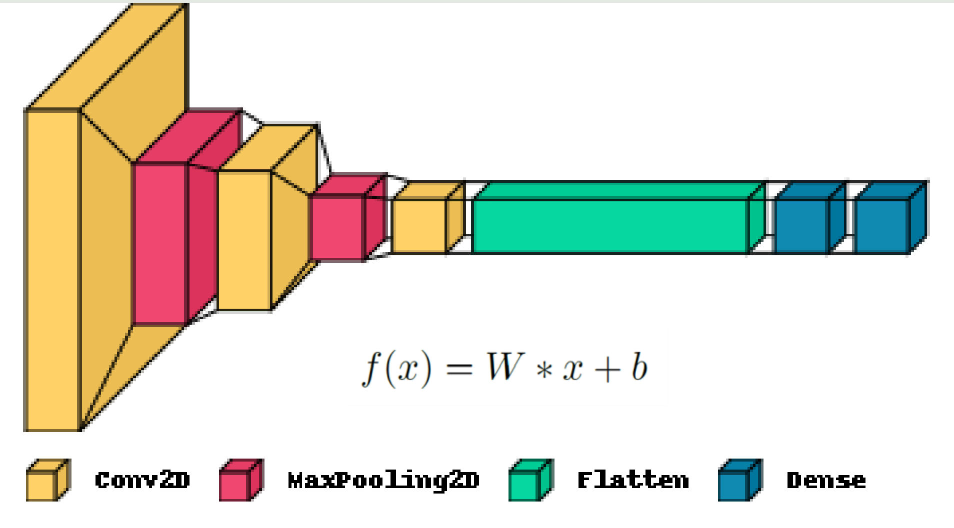


Network Architectures:

1. Basic CNN

A basic CNN is a simple feed-forward network that is designed to provide a foundation for understanding more complex models

The basic structure of a CNN for training on the dataset involves several key components: the input layer, convolutional layers, pooling layers, fully connected layers, and an output layer with an activation function.



2. ResNet

ResNet is a deep residual learning framework designed to address vanishing gradient problem by adding shortcut connections to bypass layers.

$$Y = F(x, W_i) + x$$

ResNet is known for its high accuracy and ability to handle large datasets.

We have employed the ResNet 50 model, chosen for its remarkable depth, skip connections, and advanced architecture, which collectively enhance its ability to capture intricate features within the images.

3. VGG

Visual Geometry Group. It is known for its use of small convolutional filters and deep network architecture, which helps to capture fine-grained details in images

$$f(x) = W_2 \cdot (W_1 \cdot (W_1 \cdot x + b_1) + b_2) + b_3$$

VGG differs from a basic CNN by having more layers and using smaller filter sizes of $3 * 3$ pixels. This allows VGG to learn more complex features from the input image but also makes it more computationally expensive to train and run.

To mitigate this issue, VGG typically uses max pooling layers after every two or three convolutional layers to reduce the spatial dimensions of the output.

4. MobileNet

MobileNet is a lightweight CNN architecture

$$f(x) = (W_1 * x)W_2 + b$$

MobileNet is a computationally efficient version of the CNN that uses ‘depthwise separable convolutions’ to reduce the computational complexity while maintaining accuracy.

Its lightweight architecture helps to reduce computation time and energy consumption.

5. DenseNet

DenseNet is a convolutional neural network, known for its dense connectivity pattern, where each layer is connected to all previous layers, which helps to reduce the number of parameters and mitigate overfitting.

$$f(x) = [f_1(x), f_2(x), ..., f_k(x)]$$

The dense blocks allow for a more efficient flow of information and have been shown to improve accuracy and convergence speed for image classification tasks.

Training Details: Architecture Enhancement

TABLE I
ARCHITECTURE COMPARISON OF THE LAYER STRUCTURE[1]

Basic CNN	**ResNet**	**VGG**	**MobileNet**	**DenseNet**
conv2d (Conv2D)	resnet50 (Functional)	input_2 (InputLayer)	mobilenetv2_1.00_224 (Functional	densenet201 (Functional)
max_pooling2d (MaxPooling2D)	flatten (Flatten)	block1_conv1 (Conv2D)	global_max_pooling2d_1 (GlowbalMaxPooling2D)	flatten_1 (Flatten)
conv2d_1 (Conv2D)		block1_conv2 (Conv2D)	dense (Dense)	batch_normalization_2 (BatchNormalization)
max_pooling2d_1 (MaxPooling2D)		block1_pool (MaxPooling2D)		dense_3 (Dense)
conv2d_2 (Conv2D)		block1_conv2 (Conv2D)		dropout_2 (Dropout)
flatten (Flatten)		block1_pool (MaxPooling2D)		batch_normalization_3 (BatchNormalization)
dense (Dense)		block2_conv1 (Conv2D)		dense_4 (Dense)
dense_1 (Dense)		block2_conv2 (Conv2D)		dropout_3 (Dropout)
flatten_1 (Flatten)		block2_pool (MaxPooling2D)		dense_5 (Dense)
dense_2 (Dense)		block3_conv1 (Conv2D)		
dense_3 (Dense)		block3_conv2 (Conv2D)		
		block3_conv3 (Conv2D)		
		block3_conv4 (Conv2D)		
		block3_pool (MaxPooling2D)		
		*		
		flatten_1 (Flatten)		
		dense_1 (Dense)		

Layers:

Conv2D: this layer performs a convolution operation on the input data to extract relevant features

MaxPooling2D: this layer applies max pooling to reduce the spatial dimensions of the input, preserving the most important features

Flatten: this layer reshapes the input data into a 1- dimensional vector, preparing it for the fully connected layers

Dense: these layers are fully connected layers that perform a linear transformation on the input data, followed by an activation function, to generate class predictions

Batch Normalization: this layer normalises the input data, helping with training stability and improving the learning process

Dropout: this layer randomly sets a fraction of input units to 0 during training, which helps prevent overfitting

GlobalMaxPooling2D: this layer applies max pooling across the entire feature map, reducing the spatial dimensions to a single value for each feature map.

Hyperparameter Optimisation

TABLE II
HYPER-PARAMETER TUNING

Model Name	**Basic CNN**	**ResNet**	**VGG**	**MobileNet**	**DenseNet**
Optimizer	Adam	SGD	Adam	Adam	Adam
Epochs (CIFAR11)	20	10	3	15	5
Epochs (CIFAR2)	3	3	3	5	3
Activation Function	ReLU	SoftMax	SoftMax	SoftMax	SoftMax
Batch Size	16	32	16	32	32

Specific measures were taken to address overfitting in the ResNet model: Initially, the model was trained using a learning rate of 0.001, and further fine-tuning was performed by employing SGD with a momentum value of 0.9, potentially improving its performance and convergence.

Result and Analysis: CIFAR11

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

TABLE III

COMPARATIVE MODEL SUMMARY OF 5 NETWORK ARCHITECTURES OF CIFAR-11

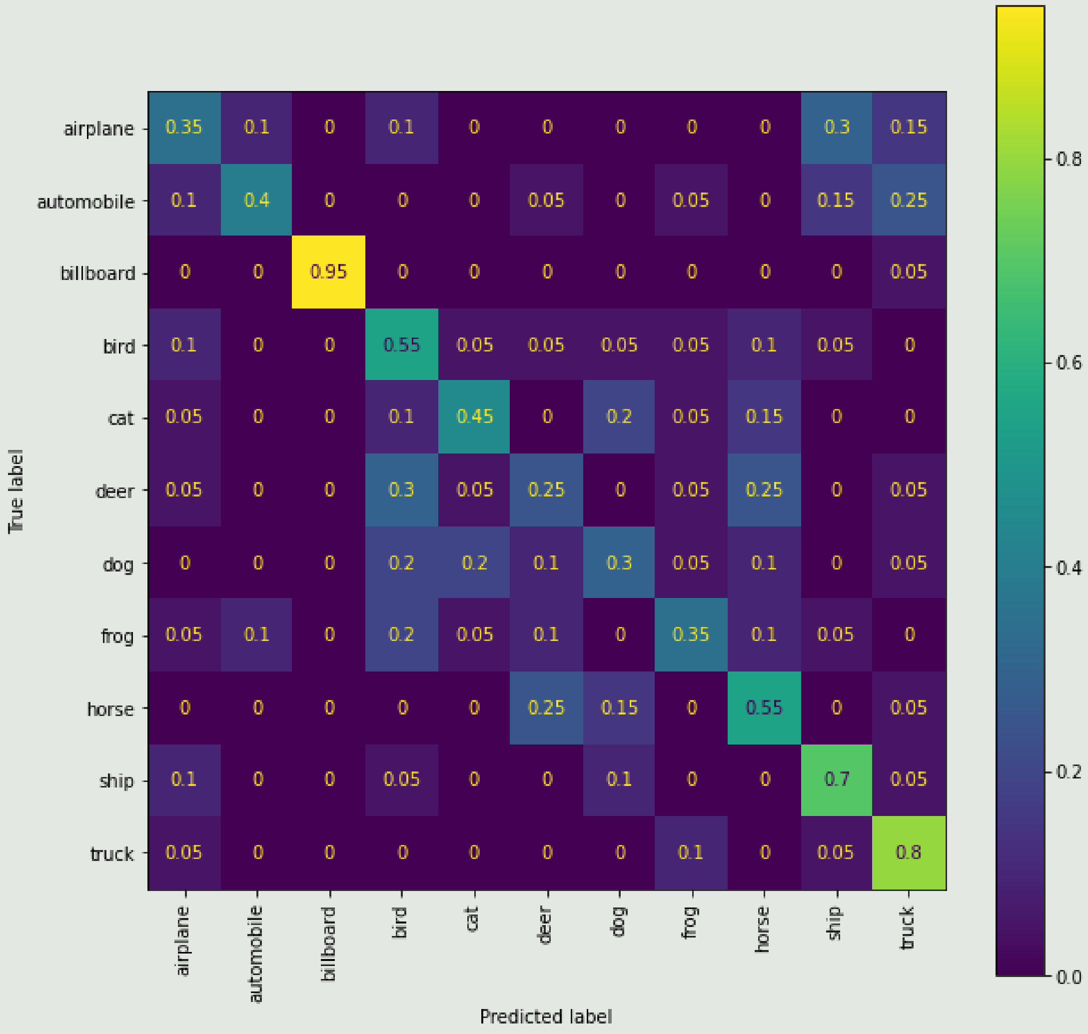
Class Name	**CNN**			**ResNet**			**VGG**			**MobileNet**			**DenseNet**		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Airplane	0.41	0.35	0.38	0.55	0.80	0.65	0.57	0.65	0.6	0.60	0.45	0.51	0.58	0.75	0.65
Automobile	0.67	0.40	0.50	0.60	0.45	0.51	0.40	0.70	0.51	0.78	0.35	0.48	0.60	0.45	0.51
Billboard	**1.00**	**0.95**	**0.97**	**0.79**	**0.95**	**0.86**	**0.86**	**0.90**	**0.88**	**1.00**	**0.05**	**0.10**	**1.00**	**0.85**	**0.92**
Bird	0.37	0.55	0.44	0.38	0.50	0.43	0.29	0.35	0.32	0.27	0.60	0.37	0.41	0.45	0.43
Cat	0.56	0.45	0.50	0.35	0.30	0.32	0.28	0.25	0.26	0.16	0.55	0.25	0.40	0.40	0.40
Deer	0.31	0.25	0.28	0.32	0.55	0.41	0.43	0.50	0.47	0.08	0.15	0.11	0.44	0.60	0.51
Dog	0.38	0.30	0.33	0.67	0.40	0.50	0.50	0.35	0.41	0.50	0.35	0.41	0.63	0.60	0.62
Frog	0.50	0.35	0.41	0.59	0.50	0.54	0.44	0.35	0.39	0.43	0.30	0.35	0.45	0.25	0.32
Horse	0.44	0.55	0.49	0.64	0.45	0.53	0.44	0.40	0.42	0.67	0.10	0.17	0.56	0.50	0.53
Ship	0.54	0.70	0.61	0.55	0.55	0.55	0.60	0.45	0.51	0.50	0.25	0.33	0.48	0.50	0.49
Truck	0.55	0.80	0.65	0.92	0.55	0.69	0.85	0.55	0.67	0.57	0.20	0.30	0.50	0.60	0.55

The 'Billboard' class consistently performs well across architectures; the performance of other classes varies. Some classes, such as 'Airplane' and 'Truck' tend to achieve relatively high scores, while others, like 'Cat' and 'Deer' show lower scores.

ResNet Confusion Matrix: CIFAR11

Using CIFAR11, the testing accuracy results:

- Basic CNN 51.36%
- ResNet 54.55%
- VGG 49.55%
- MobileNet 30.45%
- DenseNet 54.09%.



Result and Analysis: CIFAR2

TABLE IV
COMPARATIVE RESULT OF 5 NETWORK ARCHITECTURES OF CIFAR-2

Class Name	CNN			ResNet			VGG			MobileNet			DenseNet		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Billboard	0.94	0.85	**0.89**	0.51	1.00	**0.68**	0.86	0.95	**0.90**	0.83	1.00	**0.91**	0.95	1.00	**0.98**
Ship	0.86	0.95	0.90	1.00	0.05	0.10	0.94	0.85	0.89	1.00	0.80	0.89	1.00	0.95	0.97

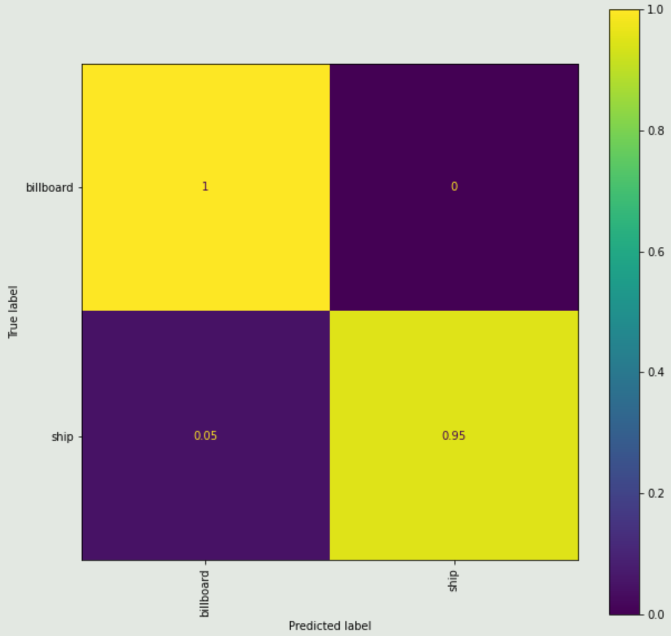


Figure: CIFAR2 DenseNet Confusion Matrix

The precision and recall values for the DenseNet architecture are close to 1.0 which indicates good performance

Conclusion

ResNet (CIFAR11) and DenseNet (CIFAR2) are strong candidates compared with the other 3 CNN for Billboard Classification.

It is important to note that the choice of architecture depends on the specific requirements of the image classification task at hand,

Computational complexity and deployment environment may also influence the final choice of architecture.

Questions?

Thank You!