Deep Learning for Billboard Classification

Sayali Avinash Chavan Dermot Kerr Sonya Coleman Hussein Khader*
Intelligent Systems Research Centre
University of Ulster
Londonderry, United Kingdom
**The Neuron*, Amman, Jordan
email:{chavan-s, d.kerr, sa.coleman}@ulster.ac.uk, hussein.khader@theneuron.com

***Abstract*—Advertising is essential to increase product awareness and foster a positive outlook, which in turn helps sales. To promote the brand and its products, billboard advertisements are widely used. This paper presents a novel approach for classifying billboards. The proposed method utilises Convolutional Neural Network (CNN) architectures to extract features from the images to enable classification. The model is trained on a dataset of billboards collected from various locations and achieves results that demonstrate high classification accuracy. The system is trained and evaluated using the CIFAR10 dataset, which includes 10 classes of objects and an additional 11th class - 'billboard', is included. The experiment utilises five different CNN architectures: Basic CNN, ResNet, Visual Geometry Group (VGG), MobileNet, and DenseNet. The performance and evaluation of each architecture are presented in detail, and extensive experiments and comparisons are conducted to determine the most effective model for classifying billboards. The results indicate that a CNN and its architectural designs are a promising solution for automating the classification of billboards in the wild.**

***Index Terms*—Classification; CNN-Architecture; Billboard; Image-Processing; CIFAR10**

I. INTRODUCTION

Out of Home (OOH) advertising is one of the most powerful channels, helping connect brands with a large and diverse audience when they are outside of their homes. This type of advertising offers several advantages including the ability to reach specific audiences using visually striking images that can increase sales and maintain brand visibility in the market. It is considered less intrusive than other forms of advertising as consumers choose whether or not to engage with it. Such advertisements can be found on roadside billboards, transit stations, street furniture, retail outlets, health and beauty, point of care and office buildings just to name a few [1]. The visibility of OOH advertisements can vary. For example, a billboard's size, shape, location and viewing angle affect visibility and in some cases, obstructions such as trees, traffic signals and vehicles can partially block an onlooker's view of a billboard. As a result, detecting billboards in images can be challenging due to the dynamic nature of their content [2] [3].

The purpose of this study is to create a classifier for billboard images using a CNN model that can accurately categorise billboards amidst other types of images in natural environments. The goal is to train and test 5 CNN architectures using billboard images merged with a subset of the CIFAR10 dataset [4], evaluate their effectiveness and determine the most efficient deep learning model for real-time billboard classification [5] [6]. The complexity lies in discerning billboards, which often blend seamlessly into the urban landscape, demanding the model's heightened perceptual acuity. Moreover, the efficacy of our proposed method in classifying billboards becomes evident when juxtaposed with classification against 10 distinct categories present in the CIFAR-10 dataset. Billboards, despite being distinct from these classes, can share similar visual characteristics with urban structures, causing intricacies in differentiation [2].

Computer vision techniques are driving a revolution across diverse applications, encompassing object and pattern recognition, precise image segmentation, robust facial detection, and even pioneering advancements in robotics and autonomous driving [7] [8] [9]. Object classification is a fundamental problem in computer vision and has been extensively studied in the machine learning and computer vision communities [10]. One of the earliest and most successful approaches to object classification is the use of hand-crafted features, such as edge detection and texture analysis [11]. These methods rely on expert knowledge and domain-specific heuristics to extract informative features from images and use them to train a classifier [12]. Some of the well-known classifiers, logistic regression, naive bayes, k-nearest neighbors, decision tree etc., were utilised for classification purposes [13]. These classifiers are widely recognised and frequently employed in various machine learning tasks due to their simplicity, interpretability, and reasonable performance. However, these methods are limited in their ability to capture complex and abstract patterns and require a significant amount of manual effort to design and implement which led to advancements in deep learning [14] [15].

Recent advances in machine learning and deep learning have led to the development of more powerful and flexible object classification algorithms [14] [16]. These methods use CNNs to automatically learn rich and complex features from raw images, without the need for manual feature engineering. CNNs have achieved state-of-the-art performance on many benchmarks [17] and are widely used in practical applications. Current studies have centred on addressing these challenges and advancing the state of the art in object classification. However, there are still many challenges and open questions in the field of object classification [15]. These include the

 ISBN: 978-1-68558-115-2

need for large amounts of labelled training data, the sensitivity of deep learning models to small perturbations in the input and the difficulty of explaining and interpreting the decisions made by these models [18]. This includes the development of CNN architectures, such as the VGG and ResNet architectures [16] [19], the use of transfer learning and other techniques to improve the efficiency and robustness of these models and the exploration of more interpretable and explainable approaches to object classification [4] [14] [20].

Hence, in this research, we investigate well-known architectures, such as ResNet, VGG, MobileNet, and DenseNet to build a billboard classifier. The study provides a detailed analysis of each architecture's performance, facilitating a comprehensive understanding of the achieved results. To illustrate our research trajectory, this paper adheres to the designated structure in the forthcoming sections. In Section 2, we delve into the Methodology, addressing dataset creation and billboard classifier design. Following that, Section 3, sheds light on Network Architectures, laying out the specific layers, training details, hyperparameter optimisation and provides comprehensive information about the five distinct network architectures employed in the research. Subsequently, Section 4 is dedicated to the Results and Analysis of the applied methods, with a focus on evaluating the performance using F1 scores on datasets CIFAR11 and CIFAR2. Lastly, the final section encapsulates the conclusion.

II. Methodology

This section discusses the dataset used and the methodology employed for the image classification experiment using CNN architectures.

A. *Dataset*

The CIFAR10 dataset [21] [22] is often used as a benchmark for evaluating the performance of different image classification algorithms. The 10 in CIFAR10 dataset represents 10 classes with 6000 images per class, each image is of size $32*32$ pixels. In total this dataset is comprised of 60,000 colour images. The 10 classes are as follows in alphabetic order: Airplane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship, and Truck.

For our experiment, we have integrated a new class denoted 'Billboard' in the dataset. This class of images was collected by the OOH industry partner. The Billboard images shown in Figure 1 were cropped and resized to $32*32$ pixels to be consistent with the CIFAR10 standard. Several observations are made for the billboard class such as images including different types of billboards, different locations, different backgrounds, obstructions caused by trees, vehicles or people, and other background clutter, etc. A total of 778 images were deemed suitable for use in the experiment. However, this created class imbalance while merging this class into CIFAR10 which has 6000 images per class.

Imbalanced datasets can negatively affect the performance of machine learning models by biasing the training towards the majority class and reducing the ability of the model to accurately classify the minority classes. Previous studies have shown that balancing the dataset can improve the performance of CNNs in image classification tasks [23]. Hence, we selected approximately the first 700 images of each of the 10 classes in the CIFAR10 dataset. By selecting an equal number of images for each class, we aimed to improve the accuracy and robustness of the CNN models used in this experiment [24]. Billboard images were pre-processed to match the CIFAR10 standards creating two separate datasets, denoted as CIFAR11 and CIFAR2 for nomenclature purposes, as briefly stated below:

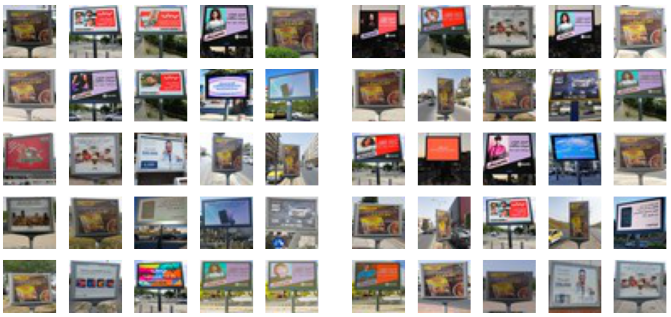


Fig. 1. Billboard class images.

- 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 images belonging to 11 classes.
- CIFAR2: Based on the outcomes in Table III, 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 images belonging to 2 classes.

B. *Billboard Classifier*

To conduct the research, five CNN architectures, namely a basic CNN, ResNet, VGG, MobileNet and DenseNet, have been utilised [16] [19] [25] [26]. Our approach involved building upon pre-trained models. The transfer learning was applied for ResNet, VGG, MobileNet, and DenseNet architectures [27].

The basic building blocks used for this experiment are shown in Figure 2. The standard methodology for image classification using CNNs involves several steps. Firstly, a dataset comprising images and their respective labels is split into training and validation sets. These images are then pre-processed by performing data normalization to make them compatible with the CNN architecture. The preprocessed data are fed into the classifier for training, which is one of the five CNN architectures used in the experiment. The classifier is trained on the validation and training datasets to extract features from the images using convolutional layers, activation functions, and pooling layers. The output is then passed through fully connected layers to classify the images into their respective classes. Backpropagation is applied to adjust the

 ISBN: 978-1-68558-115-2

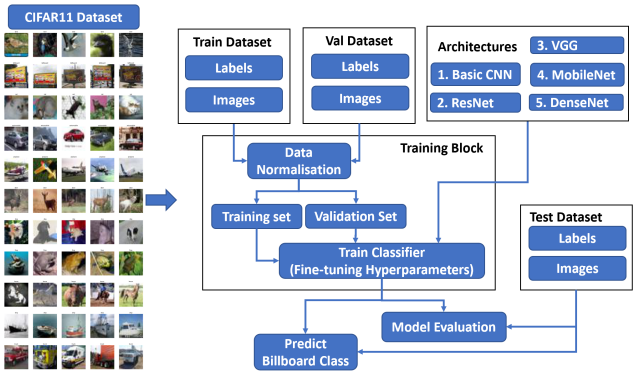


Fig. 2. Billboard classifier building blocks.

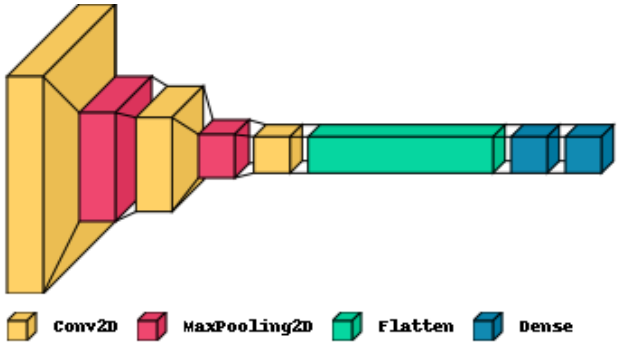


Fig. 3. Basic CNN Visualisation in Layered View.

weights and biases to train the network. Finally, the training process is repeated and fine-tuned using hyper-parameters such as a number of epochs, learning rate, batch-size, etc., to improve the accuracy. The model's performance is evaluated on an unseen test set [12].

III. NETWORK ARCHITECTURES

Each of the five CNN architectures has its own strengths and weaknesses, and the choice of architecture depends on the specific requirements of the image classification task at hand [7]. For example, ResNet [16] and VGG [19] are good choices for large, complex datasets, while MobileNet [25] is a good choice for deployment on mobile devices with limited computational resources. DenseNet [26] is a good choice for image classification tasks that require a high degree of accuracy and fast convergence.

A. *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. These components work together to learn to recognize and classify the objects in the images [28]. The convolutional layer is used for extracting features from the input image. The basic formula for a single convolutional layer can be represented as:

$$f(x) = W * x + b \tag{1}$$

where $f(x)$ is the output of the convolutional layer, W is the set of weights filters, x is the input image, and b is a bias term.

B. *ResNet*

ResNet is a deep residual learning framework that was introduced in 2015 by He et al. [16]. It is designed to address the vanishing gradient problem in deep networks by adding shortcut connections that bypass one or more layers. This allows for easier training of much deeper networks to learn residual mappings, which can help to improve performance for image classification tasks. ResNet is known for its high accuracy and ability to handle large datasets. For our experiment, 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. The output of the residual block can be represented as:

$$Y = F(x, W_i) + x \tag{2}$$

where x is the input to the residual block, $F(x, W_i)$ is the residual mapping function that is learned by the block, W_i are the learnable parameters of the block, and Y is the output of the block. The "+ x" term in the equation represents the identity mapping, which allows the output of the block to include the original input x.

C. *VGG*

Visual Geometry Group (VGG) is a CNN architecture, that was proposed in 2014 by Simonyan et al. [19]. It is known for its use of small convolutional filters and deep network architecture, which helps to capture fine-grained details in images. 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. By downsampling the output, VGG is able to decrease the computational load of the network while still retaining important features.

$$f(x) = W_2 \cdot (W_1 \cdot (W_1 \cdot x + b_1) + b_2) + b_3 \tag{3}$$

where x is the input image, W_1 and W_2 are weight matrices, b_1, b_2, and b_3 are bias vectors represents the max pooling operation.

D. *MobileNet*

MobileNet is a lightweight CNN architecture that was introduced in 2017 by Howard et al. [25]. It was designed for use in mobile and embedded devices and is known for its efficiency and accuracy. MobileNet is a computationally

 ISBN: 978-1-68558-115-2

TABLE I
ARCHITECTURE COMPARISON OF THE LAYER STRUCTURE

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)		

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, while still providing good performance for image classification tasks. MobileNet uses depthwise separable convolutions, and can be represented as:

$$f(x) = (W_1 * x)W_2 + b \tag{4}$$

where $f(x)$: Output of the MobileNet layer, W_1: Depthwise convolutional kernel, W_2: Pointwise convolutional kernel, x: Input to the MobileNet layer (e.g., an image), b: Bias term.

E. DenseNet

DenseNet is a convolutional neural network that was introduced in 2016 by Huang et al. [26]. It is 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. 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. The output of a single dense block can be represented as:

$$f(x) = [f_1(x), f_2(x), ..., f_k(x)] \tag{5}$$

where $f(x)$: Output of the DenseNet layer, $f_i(x)$: Output of the i-th dense block, x: Input to the DenseNet layer, k: Number of dense blocks in the DenseNet layer.

F. Training Details: Architecture Enhancement

Figure 3 presents the basic layer outline. In Table I, the layers and their implementations employed are explained for the following CNN architectures: Basic CNN, ResNet, VGG, MobileNet, and DenseNet. These architectures are formed by incorporating additional layers and implementations, which act as essential building blocks. They enable efficient feature extraction and effective classification tasks for the given dataset. In the context of Table I, the symbol '$*$' serves as an indicator, denoting the repetition of block 3, which occurs twice. A brief explanation of the purpose and functionality of the layers is provided below [29]:

- 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 Optimization: Table II illustrates the hyperparameters employed for training each network after utilising the default parameters. A meticulous selection and fine-tuning process were undertaken to optimise the performance of each network. The hyperparameters were carefully tuned to maximise the performance of each model. Table II presents a concise summary of the hyperparameters utilised for each model and a brief explanation for each parameter is given below:

- Epochs: The number of epochs determines the number of times the training process is repeated. After each epoch's the model's output is compared to the ground truth (actual values), and the loss function calculates the difference between them then it adjusts the weight. This newly created weight is then given to system for next epoch for training and this process goes on until the best possible accuracy is achieved without overfitting. This value is different for each model used.
- Early Stopping: The number of epochs used with the CIFAR-11 dataset differs from CIFAR-2 due to the difference in the number of images. CIFAR-11 has 7954

 ISBN: 978-1-68558-115-2

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

images, while CIFAR-2 has only 1230 images. To prevent overfitting, the dataset with fewer images (CIFAR-2) required early stopping with the number of epochs given in Table II for each network that proved to be effective in attaining optimal performance. By leveraging the early stoppage method [30], overfitting was sucessfully overcome for CIFAR-2.

- Batch Size: this was adjusted according to the availability of computational units, aiming to efficiently utilise the computational resources to benefit from parallel processing and improve training efficiency.
- Activation Function: the Softmax activation function is commonly employed in multi-class classification tasks as it converts the model's output into probability distributions across different classes used for ResNet,VGG, MobileNet and DenseNet. The Basic CNN utilised the ReLU activation function, which is a widely used activation function known for its ability to mitigate the vanishing gradient problem and introduce non-linearity to the model. Allowing the model to capture complex patterns and features essential for classification tasks.
- Optimizer: The optimizer used in the training process varied across the models. Basic CNN, VGG, MobileNet, and DenseNet were optimised using the Adam optimizer, which is an adaptive learning rate optimisation algorithm known for its efficiency in handling sparse gradients and noisy data. It computes adaptive learning rates for each parameter by taking into account the exponential decay rates of past gradients and their squared gradients. This helps the model converge faster and achieve better performance. Conversely, the ResNet utilised Stochastic Gradient Descent (SGD) as the optimizer. SGD is a widely used optimisation algorithm that iteratively updates the model's parameters based on the gradients computed on randomly selected mini batches of data [29].

In addition to the hyperparameters listed in Table II, 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. Through careful selection and fine-tuning all networks hyperparameters, the models were trained and optimised to achieve the best possible performance on the CIFAR-11 and CIFAR-2 datasets, considering the differences in the number of images and potential overfitting issues.

IV. RESULT ANALYSIS

The following section presents evaluation results for billboard classification. Tables III and IV present the results of evaluating 5 different deep learning architectures (CNN, ResNet, VGG, MobileNet, and DenseNet) on different classes of images using the CIFAR11 and CIFAR2 datasets. The performance metrics used are precision, recall, and F1-score [31]. The F1-score is a commonly used performance metric for evaluating multi-class classifiers. It is the harmonic mean of precision and recall. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Recall (or Sensitivity) is the ratio of correctly predicted positive observations to all actual positive observations in the dataset:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{6}$$

In terms of performance for the 'Billboard' class in Table III, different network architectures showed varying levels of accuracy. Among the architectures evaluated, the basic CNN, DenseNet, and VGG exhibited relatively high F1-scores of 0.97, 0.92, and 0.88, respectively. These scores suggest that these architectures achieved a high level of precision and recall in classifying images belonging to the 'Billboard' class, resulting in a balanced overall performance. On the other hand, the ResNet architecture attained an F1-score of 0.86, indicating a slightly lower but good level of accuracy for this class. However, the MobileNet architecture exhibited the lowest F1-score of 0.10 for the 'Billboard' class, implying that it struggled to accurately classify images within this specific class. The comparatively diminished performance of MobileNet might stem from its lightweight architecture optimised for efficiency, which could compromise its ability to capture intricate dataset features, unlike more complex counterparts such as ResNet, DenseNet, and VGG, which have shown consistently good results across all classes.

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. The 'Ship' class in the CIFAR-11 dataset demonstrates relatively consistent performance across the network architectures. The F1-scores for this class range from 0.48 to 0.61 suggesting a balance between precision and recall, indicating that the models achieve a reasonable trade-off between correctly identifying 'Ship' images and minimising mis-classifications. Therefore, this specific 'Ship' class with 'Billboard' has been chosen for subsequent CIFAR2 class analysis.

Using CIFAR11, the testing accuracy results for the basic CNN, ResNet, VGG, MobileNet, and DenseNet are 51.36%, 54.55%, 49.55%, 30.45%, and 54.09% respectively. It is evident that ResNet and DenseNet performed well compared to other models. These results highlight the effectiveness of the CIFAR11 model in accurately predicting the target class based on the testing data set. The highest testing accuracy achieved

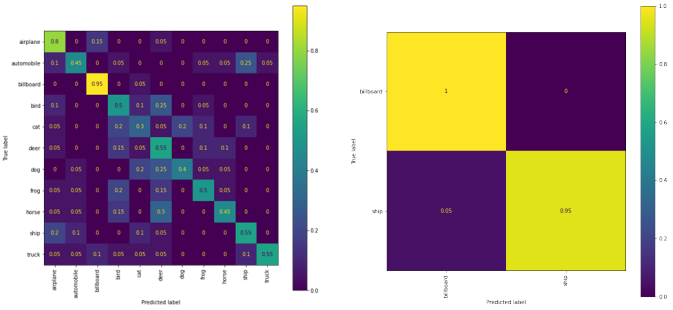
 ISBN: 978-1-68558-115-2

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

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



(a) CIFAR11 - ResNet. (b) CIFAR2 - DenseNet.

Fig. 4. Confusion Matrix.

by ResNet among all the models makes it a prime candidate for further analysis using the confusion matrix shown in Figure 4(a), which provides a visual representation of the correct and incorrect predictions for each class.

Considering CIFAR2, observing Table IV, we can see that the highest F1-score across all architectures and classes is achieved by the DenseNet with a value of 0.98 for billboard class. For the class 'Ship', as well the highest F1-score is achieved by DenseNet with a value of 0.97. It can be concluded that the DenseNet architecture outperforms other architectures in terms of the F1-Score for both the 'Billboard' and 'Ship' classes which is also reflected in Figure 4(b) using a confusion matrix for detailed model performance of each class. The precision and recall values for the DenseNet architecture are close to 1.0 which indicates good performance.

V. CONCLUSION

This paper presents deep learning based approaches for billboard classification. Based on the results presented, we can determine that ResNet (CIFAR11) and DenseNet (CIFAR2) are strong candidates compared with the other 3 CNN, particularly for the dataset for automating the classification of billboards and provide a promising solution for this application domain. However, it is important to note that the choice of architecture depends on the specific requirements of the image classification task at hand, and other factors such as computational complexity and deployment environment may also influence the final choice of architecture. Overall, the field of object classification continues to evolve and advance, with many exciting developments and opportunities for further progress.

ACKNOWLEDGMENT

We would like to express our gratitude to Digital Natives for their generosity in providing the dataset and funding the International PhD Scholarship at Ulster University.

REFERENCES

[1] T. Evans, J. Molly, S. O. Eva, and A. Miles, "The Importance of Billboard Advertising," *International Digital Organization for Scientific Research*, vol. 5, no. 1, pp. 59–65, 2020.

[2] S. Chavan, D. Kerr, S. Coleman, and H. Khader, "Billboard detection in the wild," pp. 57–64, Sep. 2021, irish Machine Vision and Image Processing Conference 2021, IMVIP2021 ; Conference date: 01-09-2021 Through 03-09-2021. [Online]. Available: https://iprcs.github.io/IMVIP.html [retrieved: Aug, 2023]

[3] C. R. Taylor, G. R. Franke, and H. K. Bang, "Use and effectiveness of billboards perspectives from selective-perception theory and retail-gravity models," *Journal of Advertising*, vol. 35, pp. 21–34, 2006.

[4] T. Ho-Phuoc, "Cifar10 to compare visual recognition performance between deep neural networks and humans," *arXiv preprint arXiv:1811.07270*, 2018.

[5] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do CIFAR-10 Classifiers Generalize to CIFAR-10?" pp. 1–25, 2018. [Online]. Available: http://arxiv.org/abs/1806.00451 [retrieved: Aug, 2023]

[6] D. Park, D. Papailiopoulos, and K. Lee, "Active Learning is a Strong Baseline for Data Subset Selection," *Conference on Neural Information Processing Systems (NeurIPS 2022)*, no. NeurIPS, pp. 1–9, 2022.

[7] J. Chai, H. Zeng, A. Li, and E. W. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios," *Machine Learning with Applications*, vol. 6, p. 100134, 2021.

[8] N. Sünderhauf et al., "The limits and potentials of deep learning for robotics," *International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 405–420, 2018.

[9] Q. Rao and J. Frtunikj, "Deep learning for self-driving cars: Chances and challenges: Extended Abstract," *Proceedings - International Conference on Software Engineering*, pp. 35–38, 2018.

 ISBN: 978-1-68558-115-2

[10] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] D. G. Lowe, "Object recognition from local scale-invariant features," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, 1999.

[12] I. Iqbal, G. A. Odesanmi, J. Wang, and L. Liu, "Comparative Investigation of Learning Algorithms for Image Classification with Small Dataset," *Applied Artificial Intelligence*, vol. 35, no. 10, pp. 697–716, 2021. [Online]. Available: https://doi.org/10.1080/08839514.2021.1922841 [retrieved: Aug, 2023]

[13] S. Khan et al., "Comparison of multiclass classification techniques using dry bean dataset," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 6–20, 2023.

[14] H. Jiang et al., "A review of deep learning-based multiple-lesion recognition from medical images: classification, detection and segmentation," *Computers in Biology and Medicine*, vol. 157, p. 106726, 2023. [Online]. Available: https://doi.org/10.1016/j.compbiomed.2023.106726 [retrieved: Aug, 2023]

[15] F. Sultana, A. Sufian, and P. Dutta, "Advancements in image classification using convolutional neural network," *Proceedings - 2018 4th IEEE International Conference on Research in Computational Intelligence and Communication Networks, ICRCICN 2018*, pp. 122–129, 2018.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 770–778, 2016.

[17] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11263-015-0816-y [retrieved: Aug, 2023]

[18] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," *34th International Conference on Machine Learning, ICML 2017*, vol. 7, pp. 4844–4866, 2017.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14, 2015.

[20] J. Fan, J. H. Lee, and Y. Lee, "Application of transfer learning for image classification on dataset with not mutually exclusive classes," in *2021 36th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, 2021, pp. 1–4.

[21] A. Krizhevsky, "Cifar-10 and cifar-100 datasets," 2009. [Online]. Available: https://www.cs.toronto.edu/ kriz/cifar.html [retrieved: Aug, 2023]

[22] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[23] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0192-5 [retrieved: Aug, 2023]

[24] V. S. Spelmen and R. Porkodi, "A review on handling imbalanced data," in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, 2018, pp. 1–11.

[25] G. A. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017. [Online]. Available: http://arxiv.org/abs/1704.04861 [retrieved: Aug, 2023]

[26] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 2261–2269, 2017.

[27] M. Iman, H. R. Arabnia, and K. Rasheed, "A Review of Deep Transfer Learning and Recent Advancements," *Technologies*, vol. 11, no. 2, pp. 1–14, 2023.

[28] D. R. Sarvamangala and R. V. Kulkarni, "Convolutional neural networks in medical image understanding: a survey," *Evolutionary Intelligence*, vol. 15, no. 1, pp. 1–22, 2022. [Online]. Available: https://doi.org/10.1007/s12065-020-00540-3 [retrieved: Aug, 2023]

[29] L. Alzubaidi et al., "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data 2021 8:1*, vol. 8, pp. 1–74, 3 2021. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00444-8 [retrieved: Aug, 2023]

[30] Y. Wei, F. Yang, M. J. Wainwright, and S. Member, "Early stopping for kernel boosting algorithms: A general analysis with localized complexities," *IEEE Transactions on Information Theory*, vol. 65, p. 6685, 2019.

[31] N. W. S. Wardhani, M. Y. Rochayani, A. Iriany, A. D. Sulistyono, and P. Lestantyo, "Cross-validation Metrics for Evaluating Classification Performance on Imbalanced Data," *2019 International Conference on Computer, Control, Informatics and its Applications: Emerging Trends in Big Data and Artificial Intelligence, IC3INA 2019*, pp. 14–18, 2019.

 ISBN: 978-1-68558-115-2