International Journal of Computing and Digital Systems
2024, VOL. 17, NO.1, 1-12

http://dx.doi.org/10.12785/ijcds/1571031764

Classification of Road Features Using Convolutional Neural Network (CNN) and Transfer Learning

Mustafa M. Abd Zaid[1,2], Ahmed Abed Mohammed[1,2] and Putra Sumari[3]

[1]*College of Technical Engineering, Islamic University, Najaf, Iraq*
[2]*College of Technical Engineering, Islamic University, Dewania, Iraq*
[3]*School of Computer Science, University Sains Malaysia, 11800 Peneng, Malaysia*

Received 19 May 2024, Revised 18 July 2024, Accepted 4 September 2024

Abstract: Efficient and accurate classification of road features, such as crosswalks, intersections, overpasses, and roundabouts, is crucial for enhancing road safety and optimizing traffic management. This study proposes a novel classification approach that leverages transfer learning and convolutional neural networks (CNNs) to address the road feature classification problem. Our system aims to achieve robust and real-time classification of road features by employing state-of-the-art CNN architectures. The dataset comprises 7,616 images, including those from the MLRSNet dataset, with a fixed size of 256x256 pixels. After doing all the necessary pre-processing, we manually extracted satellite images from Malaysia using Google Earth Pro and merged them with MLRSNet. We designed a CNN architecture featuring 24 convolutional layers and eight fully connected layers. We also used transfer learning models such as ResNet50, MobileNetV2, VGG19, and InceptionV3. The best-performing model during the validation phase is InceptionV3, achieving an accuracy of 98.9%. In contrast, ResNet50 and VGG-19 excelled during the test phase with an accuracy of 98.7%. The proposed CNN model achieved 95.1% and 94.4% accuracy during the validation and test stages. These results underscore the effectiveness of our models in improving road feature classification. It is crucial for developing autonomous driving and traffic management systems, which contribute to the progress of intelligent transportation systems and improve road safety and efficiency.

Keywords: Road Feature Classification, CNN, Transfer Learning, MLRSNet, Hyperparameter Tuning.

1. INTRODUCTION

The rapid growth of computer vision and deep learning has significantly impacted the automated extraction and classification of road parameters. Effective classification of road features is essential for applications in transportation systems, such as autonomous driving, traffic management, and road infrastructure maintenance [1]. With ongoing urbanization, road networks are becoming increasingly complex, highlighting the need for precise identification and classification of various road elements such as crosswalks, junctions, overpasses, and roundabouts [2]. These elements provide crucial information necessary for safe navigation and compliance with traffic rules [3]. Precise categorization of road characteristics not only enhances the effectiveness of advanced driver-assistance systems (ADAS) but also supports the development of intelligent transportation systems (ITS). For instance, identifying crosswalks can notify vehicles of pedestrian presence, thereby reducing the likelihood of accidents. Similarly, recognizing junctions and roundabouts is crucial for optimizing traffic flow and minimizing congestion [4]. As the focus on autonomous vehicles intensifies, the demand for accurate road feature categorization continues to grow, as these systems rely on precise data to make informed decisions, and misclassification can lead to dangerous situations. Combining Transfer Learning and Convolutional Neural Networks (CNNs) emerges as a robust approach to improving the speed and accuracy of road feature classification.

Traditionally, road feature classification has relied on labor-intensive, subjective, and error-prone manual examination and interpretation by human experts [5]. However, advancements in computer vision, machine learning, and deep learning have paved the way for automated classification systems that can perform this task more efficiently and accurately [6]. Recent years have seen various strategies to automate road feature classification [7]. These methods employ computer vision techniques and machine learning models, particularly CNNs, to analyze and categorize images of road features. CNNs are highly effective in image recognition tasks due to their ability to extract relevant information from input images and make accurate predictions.

This paper aims to address the classification challenges

E-mail address: mustafa.majed@qu.edu.iq, a.alsherbe@gmail.com, putras@usm.my

of these unique road features by developing a CNN-based architecture and employing a transfer learning approach to identify the best-performing model [8]. By leveraging CNNs, which can autonomously learn feature hierarchies from input images, this research aims to improve performance in image processing applications. Moreover, computer vision applications like image segmentation and classification have proven highly efficient and successful using CNNs. By analyzing high-resolution remote sensing images or ground-level photographs, CNNs can recognize various road types, markings, signs, and other features, making them highly useful for classifying road attributes.

Transfer Learning, a machine learning technique that utilizes pre-trained models to build new models for different tasks, is pivotal in this approach. By leveraging knowledge from pre-trained models trained on large datasets like ImageNet, transfer learning enables the extraction of essential features from road images, simplifying the classification process. In fields like road feature recognition, where large labeled datasets are scarce, transfer learning becomes invaluable. The combination of CNNs with transfer learning captures the best features of both techniques, allowing for fine-tuning of pre-trained CNN models with minimal road-specific data, leading to improved classification accuracy and faster training.

This study contributes to the existing body of knowledge by demonstrating that combining CNNs with transfer learning provides a superior solution for road feature classification, particularly when data is limited. Furthermore, this approach enhances the model's generalizability, making it more effective with unseen data. Studies have shown that this method has yielded significant success in segmenting road types (e.g., highways, urban streets, and country roads) and road conditions (e.g., wet, dry, and icy) from aerial and ground-level images. Additionally, to further improve robustness and accuracy, CNNs can be trained using multi-modal inputs, such as combining visual images with LiDAR data.

In summary, combining transfer learning with CNNs offers a highly effective and efficient approach to road feature classification, addressing data scarcity while achieving high accuracy. This holistic strategy is crucial for intelligent transport systems and innovative city applications, enabling more efficient and effective detection and classification of road features. The advancements in road feature categorization achieved through this research will align with the ongoing technological transformation, leading to substantial benefits in the field. Implementing this research can assist traffic management authorities in automating the monitoring and control of transportation facilities, enhancing road safety, and optimizing traffic flow. Additionally, the proposed system can serve as a foundation for developing intelligent transportation systems and innovative city initiatives, ultimately improving urban mobility.

2. Related Works

In recent years, there has been much interest in classifying road characteristics, including roundabouts, crosswalks, overpasses, and intersections, utilizing transfer learning and convolutional neural networks (CNNs). Researchers have investigated several methods to improve the reliability and accuracy of classifying road features in the context of intelligent transportation systems.

Tu men et al. proposed deep learning and image processing techniques to detect intersections and crosswalks. They designed a multi-scale CNN architecture called the RoIC-CNN that incorporated convolutional and pooling layers to capture spatial information at different scales.One of the strengths of this work lies in the innovative multi-scale design, RoIC-CNN consists of ten convolution layers and eight fully connected layers.Nevertheless, there are some constraints in their methodology. Despite its effectiveness, the RoIC-CNN is a complicated model consisting of 10 convolution layers and eight fully linked layers, resulting in significant computing expenses.This study also tests other CNN models, such as VggNet-5, LeNet, and AlexNet, to compare their performance. From the evaluation, the best-performing model in detecting crosswalks and inter-sections is the RoIC-CNN [9].

Using data from six sample Chinese cities' OpenStreetMap (OSM) road networks, Li et al. constructed the flyover labeling geodatabase (OLGDB) by extending the target detection model (Faster-RCNN). A region proposal network (RPN) determines the optimal location for a flypast after convolutional neural networks (CNNs) trained on raster data acquire task-adaptive features. ZF-net, VGG-16, and Inception-ResNet V2 are the three convolutional neural networks (CNNs) that comprise Faster-RCNN. Following their synthesis into image bands for training data improvement, the five geometric metrics area, perimeter, squareness, circularity, and W/L are evaluated for their contribution to the flyover identification task, An outstanding advantage of this work is the novel integration of geometric measurements (area, perimeter, squareness, circularity, and W/L) with convolutional neural network (CNN) results.An inherent limitation of this study is its exclusive emphasis on flyovers, which may restrict the model's applicability to other crucial road characteristics. Here fine-tuning establishes the ideal learning rate and batch size combination. The experimental results show that the proposed strategy gets an acceptable level of accuracy (about 90%) [10].

Another active area of research aimed at leveraging artificial intelligence techniques to improve transportation infrastructure and operations is the analysis of the deep learning-based classification of transportation facilities for enhanced road safety and traffic management. In the study conducted by Jilani et al. (2022), a five-layer CNN deep learning model is suggested for traffic congestion categorization. Augmentation with GANs improves the traffic congestion dataset. The study used pre-trained RsNet50 and DenseNet-121 as the benchmark to compare with the 5-layer CNN. The study found that the proposed CNN emerged as the best model with an accuracy of 98.63% compared to ResNet50 (90.59%) and DenseNet-121 (93.15%), respectively [11].

This study presents a hybrid model that enables road feature identification by merging Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). The main objective is enhancing road safety and optimizing the driving experience. The model employs Convolutional Neural Networks (CNNs) based on the MobileNetV2 architecture and Recurrent Neural Networks (RNNs) to handle GRU. A benefit of using it in automobiles is its ability to process real-time data and effortlessly transfer models due to its lightweight form. The CNN-MobileNetV2-GRU model has improved its ability to accurately identify road components, including speed bumps and variations in road conditions. The model provides a comprehensive solution for classifying road features, including spatial, efficiency, and speed requirements, by processing data from the edge to the cloud. The CNN- MobileNetV2-GRU model is well-suited for practical applications such as enhanced driver assistance systems and autonomous vehicles because of its exceptional precision and efficient utilization of processing resources [12].

The authors present a novel approach for classifying roads into four distinct categories: highways, city roads, undeveloped areas, and housing estates. They introduce a sophisticated pyramidal residual network, specifically the 1D-PyramidNet model, which demonstrated the highest accuracy (92.23%) in interpreting the data. This model outperformed other deep learning models, showcasing its effectiveness in handling the complexity of road classification tasks. One of the key strengths of this study lies in its innovative use of the pyramidal residual network architecture, which leverages multi-scale feature extraction to capture nuanced patterns in the data. Additionally, the use of a public benchmark dataset alongside sensor data obtained via intelligent eyewear adds a unique dimension to the research, combining traditional and emerging data sources for more robust road classification [13].

As detecting road damage is essential to maintaining optimal road conditions and enhancing transportation safety, several studies have used object detection/classification models and a deep learning technique. Road damage algorithms can be divided into two large groups; the first group includes algorithms with two stages, and the second consists of one-stage ones. A two-state solution, like support vector machines and convolutional neural networks, includes a detector and a classifier, first defining regions in an image where an object may be present and then classifying each area. Another type, a one-stage solution, for instance, YOLO and SSD, tries to make predictions for the best possible region simultaneously, negating the need for additional steps. This work describes the development of such algorithms and demonstrates their use in explaining road damage detentions [14].

Applications reliant on vision, such autonomous driving and traffic monitoring, are already using deep learning algorithms. Intelligent transport systems primarily focus on semantic road identification and traffic sign recognition to prioritize safety. The development of intelligent transport systems is greatly impacted by these concerns. A driving assistance system that uses deep learning components is introduced in this study. utilizing transfer learning approaches, the system is built utilizing hybrid 2D-3D convolutional neural network (CNN) models. For simplicity and speed, the models use a pre-trained deep 2D Convolutional Neural Network (CNN) in conjunction with a simpler 3D CNN. As a well-established approach to the issue of traffic sign recognition, Hybrid-TSR is the first model to be considered. In order to identify road space, the second model, Hybrid-SRD, employs up-sampling and deconvolutional procedures to examine the semantic data. The proposed methods considerably enhance accuracy and efficiency, according to the predicted outcomes [15].

Autonomous cars are crucial for traffic moving monitoring, and the capability of instantly detecting potholes is essential for the safety and convenience of the vehicles. Many methods, such as reporting to authorities, vibration-based sensors, and 3D laser imaging, are limited by the high costs incurred in their installation and the possible dangers associated with their use. This article presents the new method, Adaptive Mutation and Dipper Throated Optimisation (AMDTO), which is designed to select and optimize the features of the Random Forest (RF) classifier. The AMDTO+RF technique that was employed had a pothole classification accuracy of 99. The method's effectiveness in experiment A was 795%, beyond previous methods, such as WOA+RF, GWO+RF, PSO+RF, and transfer learning approaches. The in-depth statistical analysis of the recorded outcomes proves the method's importance and consistency. This method aims to electronify the precise and quick spot identification process [16].

Automated data collecting for roadside barriers has been developed by the Wyoming Department of Transportation (WYDOT) as part of its asset management system. The system collects the geometric attributes and material conditions of barriers, which in turn assists in asset management decision-making and, thus, optimization. There are over one million linear feet of state barriers, totalling over $100 million. The price for the state to acquire these features is more than half a million dollars at once. A unique method was suggested to identify different kinds of roadside barriers by using pre-trained models like inception v3, denseness 121, and VGG 19. VGG 19 network was used, which resulted in an excellent accuracy of 97% through transfer learning. An architectural non-transfer model, which is a model that is built and is very simple, was made, and the accuracy of the model was 85%. On the other hand, the non-transfer learning model was better than the inception and denseness models but still needs to be better than the VGG network [17].

The study introduces a technique for classifying road signs using pre-trained Convolutional Neural Network (CNN) models based on transfer learning. The authors assess the efficacy of their models using the German Traffic Sign Recognition Benchmark test dataset. The researchers use transfer learning and augmentation approaches to assist in different designs. The findings demonstrate that the suggested strategy attains an average accuracy of 99.2%, surpassing the performance of current approaches. These findings indicate that using transfer learning and pre-trained models may significantly improve the accuracy of road sign categorization, even when working with a limited dataset [18].

The effectiveness of pre-trained networks based on deep learning in evaluating gravel road pictures using conventional methods is tested in this study. Images from Google Street View and self-recorded videos make up the collection. These photos have been hand-tagged according to standards set by Sweden's Road Maintenance Agency. The training and testing datasets were split 60:40. A number of pre-trained models were used, and they all performed well, with an accuracy rate above 92%. In terms of accuracy and F1 score performance, the pre-trained VGG-16 model that made use of transfer learning outperformed the other models that were recommended. The study's overarching goal is to help road maintenance agencies better assess loose gravel [19].

There is an increasing need to identify wet road surfaces to address accidents and traffic problems during rainy weather conditions. Acoustic signals have garnered interest because of their cost-effectiveness in deployment. A large quantity of training data is required by current deep-learning methods, which rely on supervised audio measurements. The evolution of convolutional neural networks (CNNs) has made it easier to train CNNs on one dataset and then apply them to another. The reliability of convolutional neural network (CNN) models that have been pre-trained to detect wet road surfaces is tested in this study. The results show that transfer learning can distinguish between dry and moist surfaces with an accuracy rate of more than 80% [20].

Our technique differs by using a unique categorization framework that combines transfer learning with a specifically constructed CNN architecture. We may get excellent accuracy rates using advanced models like InceptionV3, ResNet50, and MobileNetV2 while minimizing the need for extensive labelled datasets. In addition, our approach integrates data augmentation approaches to enhance the model's resilience against fluctuations in illumination and ambient circumstances.

In addition, our work broadens the dataset by using both the MLRSNet dataset and manually extracted satellite photos, resulting in a more varied training set that improves the model's ability to generalize. Our approach, which incorporates advanced architectures, transfer learning, and a comprehensive dataset, represents a substantial advancement compared to previous methodologies. This ultimately leads to a more accurate classification of road features, which is crucial for developing autonomous driving technologies; although prior research have shown the efficacy of Convolutional Neural Networks (CNNs) and transfer learning in many situations, our method aims to expand on this past knowledge by particularly focusing on the categorization of road characteristics in various settings. The following section delineates the approach used to accomplish this.

3. Methodology

The methodology consists of a set of phases: dataset collection, preprocessing, classification models, performance metrics, and hyperparameter tuning, as shown in Figure 1.

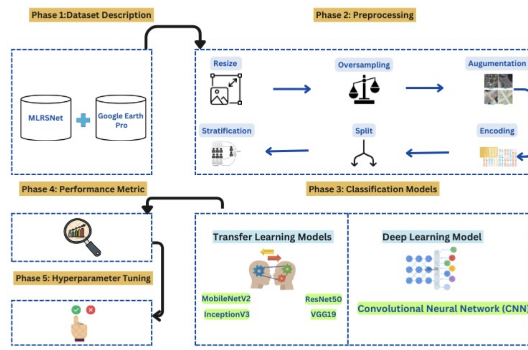

Figure 1. Methodology of the study.

A. Phase 1:Dataset Description

We will explain the dataset used in our study.

1) MLRSNet Dataset (First Dataset)

The dataset used in this study is primarily sourced from the MLRSNet dataset on the GitHub website, which is available from Mendeley Data [21]. The dataset contains 109,621 high spatial resolution optical images of 46 different categories captured from satellites. This study obtained images of roundabouts, intersections, and overpasses from the MLRSNet dataset. 2,040 roundabouts , 2,498 Intersections and 2500 overpasses images were collected from the MLRSNet dataset, respectively; we have to mention that this dataset does not have crosswalk Images. The images from the MLRSNet dataset have a fixed size of 256x256 pixels, as shown in Figure 2.

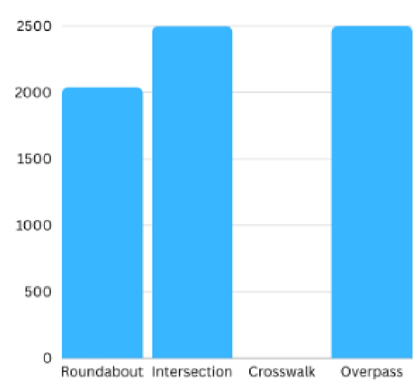

Figure 2. MLRSNet Dataset

2) Dataset from Google Earth Pro (Second Dataset)

It provides users access to a wealth of geographical data, including satellite photos, maps, topography, and 3D buildings, all from the comfort of their virtual globe [22]. Additional satellite images of roundabouts, intersections, crosswalks, and overpasses in Malaysia were obtained by taking screenshots of those features using Google Earth Pro software that contain 100, 101, 242 and 135 Roundabout, Intersections, Crosswalk and Overpass as shown in Figure 3.

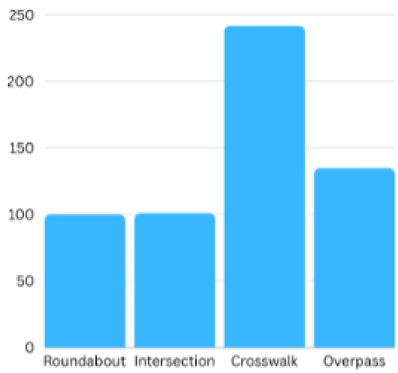

Figure 3. Dataset from Google Earth Pro

3) Merge Dataset

We merged two datasets, the MLRSNet dataset and the dataset from Google Earth Pro, to get 2140, 2599, 242 and 2635 to Roundabout, Intersection, Crosswalk and Overpass as Figure 4. and Table I, As can be observed from Table I and Figure 4, the data could be more balanced, where the number of images of crosswalks is significantly less than the images of other categories since the images of crosswalks are not explicitly available to be downloaded from the MLRSNet Dataset. Figure 5 shows the sample images for the roundabout, Intersection, crosswalk, and overpass.

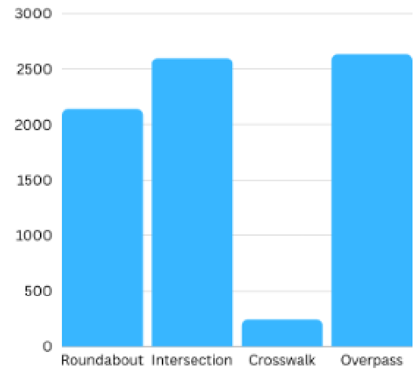

Figure 4. Merge Dataset

Figure 5. Sample of Images

B. Phase 2: Preprocessing

We apply a set of processes before entering the classification model

1) Resizing Image

It is essential preprocessing to change the size of the Image and its dimensions while maintaining its aspect ratio or stretching it to fit a new size. To ensure uniformity and compatibility, all photos in our dataset were uniformly downsized to a resolution of 224x224 pixels. The raw images collected from Google Earth Pro were of a different size and aspect ratio than the MLRSNet dataset. Thus, the images obtained from Google Earth Pro were cropped into perfect squares. This was realized through checking the aspect ratio of the images. No cropping action will be performed if the image width and height are equal. If the width exceeds the height, the left and right sides of the Image will be cropped, and a perfectly square image will be returned. If the width is less than the height, the top and bottom parts of the Image will be cropped, and a square image will be returned. The cropped images will then be resized to a resolution of 224×224. Similarly, images obtained from the MLRSNet dataset were resized to 224x224 pixels as well.

2) Oversampling

It is a method for dealing with datasets that have an imbalance between classes, where one class has a much smaller number of instances than the others. Biassed models that fail to represent minority groups adequately may result from this imbalance; class imbalance is detrimental towards training on classifiers, and it affects the convergence of

TABLE I. Number of Image to each Class

Images	MLRSNet	Google Earth Pro	Total
Roundabout	2040	100	2140
Intersection	2498	101	2599
Crosswalk	0	242	242
Overpass	2500	135	2635

the deep learning model during the training phase and generalization of the model during the test phase [23], [24]. Good model results can be attained if all the classes in the classifier are properly represented [25]. The oversampling method was among the best methods in alleviating class imbalance problems for CNN-related model training. In order to alleviate the class imbalance issue in this study, image augmentation, a form of the oversampling method, was performed so that class balance was achieved. The argumentation library from Python was used to perform image augmentation. In this data augmentation process, a random image was selected for each image class, and random augmentation operations were performed. The following augmentation techniques, with a 0.5 probability chance of execution, were performed in Figure 6

- Random rotate-90°
- Vertical flip
- Horizontal flip
- Random brightness contrast
- Random gamma

3) Image Augmentation

In order to increase the size of the dataset, we used methods for augmenting images. Incorporating data variances via image augmentation is a common way to improve the general-isolation Model's performance [26]. Data augmentation increases the number of images for each category to 2700 images, as shown in Figure 7.

4) Encoding

It is a method to change the way values are represented. When dealing with categorical variables, label encoding, also called ordinal Encoding, gives each category in the dataset a distinct integer value [27]. The input and filter weights are determined by doing the dot product using this technique, which enables the conversion of convolutional filters. The network can analyse the input image and extract relevant features and spatial information through these procedures [28].

5) Split Data

The images are then split to train, validate, and test set with a ratio of 7:2:1. Here we have the train set, which is used to train the model; the validation set, which is used to validate the model after each epoch, and the test set, which is used to assess the model after training.

Figure 6. Some of Edits on Image

6) Stratification

It is a method for ensuring that various data subsets keep the same distribution of classes in the dataset. Because biassed models might result from an uneven class distribution, this is especially crucial for classification tasks, which were performed during the split to ensure that the number of images for every class was the same for each batch of dataset.

C. Phase 3: Classification Models

We will explain all classification models that we used in our study

1) Convolutional Neural Network (CNN):

Among artificial neural networks, Convolutional Neural Networks (CNNs) stand out due to their deep feed-forward design [29]. Image processing and analysis are two areas where Convolutional Neural Networks (CNNs) shine. In order to assess the input picture, a Convolutional Neural Network (CNN) usually uses convolutional layers that use sets of adaptive filters. CNN is well-known in image-based

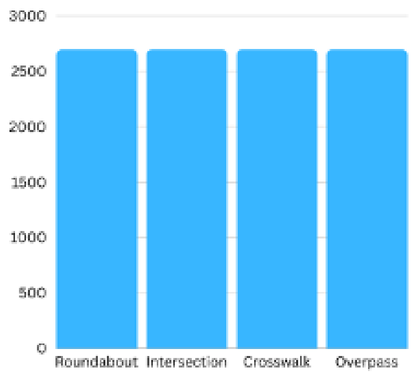

Figure 7. Number of Image after Image Augumentation

classification tasks. CNNs process the input data using several interconnected layers. Typically, a convolutional layer is the first hidden layer of a convolutional neural network (CNN). It uses a series of filters to identify patterns in the input data [30]. These networks are built to automatically extract significant features at various degrees of abstraction from raw pixel input and learn hierarchical representations. Convolutional layers in CNNs allow for the extraction of regional patterns and structures. In contrast, pooling layers make it easier to downsample spatial data, which improves the model's ability to identify essential features. Based on the retrieved features, fully connected layers provide the final categorization.

A custom CNN model is created in this study to perform the classification of roundabouts, crosswalks, intersections, and overpasses. The CNN model is created using TensorFlow. The CNN architecture consists of 24 convolution layers and eight fully connected layers; the CNN architecture comprises convolutional layers, pooling layers, normalization layers, and dropout layers. Fully connected layers follow these. As the data flows through the network, the output shapes decrease, indicating a reduction in spatial dimensions and an increase in depth (number of channels). The number of parameters varies greatly across layers, with convolutional layers having many parameters due to the learnable filters, as listed in Table II and Figure 8.

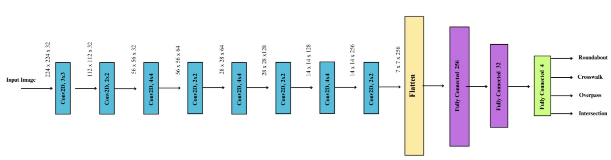

Figure 8. Diagram of CNN Custom

2) *Transfer Lerning*

In addition to the proposed CNN, transfer learning models such as ResNet50, MobileNetV2, VGG19 and InceptionV3 are used to compare the accuracy with our CNN model developed. It is a powerful technique in computer vision tasks, enabling models to leverage knowledge learned from pre-trained models trained on large-scale datasets [31]. By transferring this knowledge to a new task, transfer learning can significantly enhance classification performance. In our approach, we harness the benefits of transfer learning by utilizing pre-trained models from the literature, such as ResNet50, MobileNetV2, VGG19 and InceptionV3. In this study, the classification layers of the transfer learning models were dropped and replaced with a new classification layer similar to the proposed CNN model. The weights are initialized based on the models' weights trained on ImageNet. All the transfer learning layers were allowed to be trained in this study. Table III.. summarizes the common hyperparameters configured for the transfer learning models.

a) ResNet50

It is a 50-layer deep convolutional neural network design that uses residual connections to support the training of intense networks. It has demonstrated impressive performance in image classification challenges, overcoming the vanishing gradient issue and facilitating more straightforward deep model optimization [32].

Each Residual Block consists of two or three convolutional layers and a shortcut link that adds the input of the block to its output. Batch normalization is a technique that aids in alleviating the issue of the disappearing gradient. Global average pooling is used after each convolutional layer to enhance training stability and speed.ResNet50 employs global average pooling instead of fully linked layers, effectively decreasing the number of parameters and mitigating overfitting.

b) MobileNetV2

A compact convolutional neural network architecture called MobileNetV2 was created for effective mobile and embedded vision applications. It is appropriate for devices with limited resources because it uses depth-wise separable convolutions and inverted residual blocks to simplify computations while retaining competitive accuracy [33].

Residuals that are reversed or flipped MobileNetV2 have inverted residual blocks that use linear bottlenecks, enabling effective feature extraction with little computing expense and a lightweight architecture. The architecture is designed to maximize efficiency on devices with limited computing resources, making it well-suited for mobile applications. Additionally, it can accommodate multiple input sizes, enabling flexibility for diverse applications.

c) VGG19

Convolutional neural network architecture VGG19 is renowned for its efficiency and simplicity. There are 19 layers, including several 3×3 convolutional layers followed by max-pooling layers. Although VGG19 has more parameters than other architectures, it performs image classification tasks with high accuracy [34].

The VGG19 model has a complex architecture with many parameters, enabling it to learn intricate features. It employs a uniform structure by using small convolutional filters (3x3) throughout the network, ensuring a consistent

TABLE II. The CNN architecture

Layer	Layer Name	Output Shape	Param #
1	Conv2D	224 x 224 x 32	896
2	MaxPooling2D	112 x 112 x 32	0
3	Conv2D	112 x 112 x 32	9248
4	BatchNormalization	112 x 112 x 32	128
5	MaxPooling2D	56 x 56 x 32	0
6	Dropout	56 x 56 x 32	0
7	Conv2D	56 x 56 x 64	18496
8	BatchNormalization	56 x 56 x 64	256
9	Conv2D	56 x 56 x 64	36928
10	BatchNormalization	56 x 56 x 64	256
11	MaxPooling2D	28 x 28 x 64	0
12	Dropout	28 x 28 x 64	0
13	Conv2D	28 x 28 x 128	73856
14	BatchNormalization	28 x 28 x 128	512
15	Conv2D	28 x 28 x 128	147584
16	BatchNormalization	28 x 28 x 128	512
17	MaxPooling2D	14 x 14 x 128	0
18	Dropout	14 x 14 x 128	0
19	Conv2D	14 x 14 x 256	295168
20	BatchNormalization	14 x 14 x 256	1024
21	Conv2D	14 x 14 x 256	590080
22	BatchNormalization	14 x 14 x 256	1024
23	MaxPooling2D	7 x 7 x 256	0
24	Dropout	7 x 7 x 256	0
25	Flatten	12544	0
26	Dense	256	3211520
27	BatchNormalization	256	1024
28	Dropout	256	0
29	Dense	32	8224
30	BatchNormalization	32	128
31	Dropout	32	0
32	Dense	4	132

TABLE III. The common hyperparameters configured for the transfer learning

Parameter	InceptionV3	ResNet50	VGG-19	MobileNetV2
Input image	(224,224,3)	(224,224,3)	(224,224,3)	(224,224,3)
Weight	Initialized to ImageNet	Initialized to ImageNet	Initialized to ImageNet	Initialized to ImageNet
Optimizer	Adam	Adam	Adam	Adam
Loss function	Sparse categorical cross entropy	Sparse categorical cross entropy	Sparse categorical cross entropy	Sparse categorical cross entropy
Classifier	Softmax	Softmax	Softmax	Softmax
Epochs	50	50	50	20
Dropout rate	0.2	0.2	0.2	0.2

approach to feature extraction. The final layers of the model consist of fully connected layers that output class probabilities.

d) InceptionV3

The deep convolutional neural network architecture known as InceptionV3 uses inception modules. These modules use parallel convolutional layers with various kernel sizes to capture features at various scales. InceptionV3 reduces the number of parameters using dimensionality reduction techniques to obtain high accuracy in image recognition tasks while retaining computing efficiency [35].

The Inception modules contain many parallel convolutional layers, each with varying filter sizes and pooling layers. Factorized convolutions allow the model to acquire a diverse range of information. InceptionV3 utilizes factorized convolutions, such as splitting 3x3 convolutions into two separate 1x3 and 3x1 convolutions, in order to decrease computational complexity. Additionally, the model incorporates auxiliary classifiers during training to assist with gradient flow and enhance convergence.

D. Performance Metric

We test the CNN models in this stage to see how well they do. Accuracy is a standard metric to gauge how well a Convolutional Neural Network (CNN) is doing. Accuracy measures how many predictions were right relative to the total number of photos. Equation 1 is the accuracy formula:

$$Acc(\%) = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \quad (1)$$

True positives (TPs) occur when the model accurately predicts that a sample belongs to a given class. In contrast, true negatives (TNs) occur when the model accurately predicts that a sample does not belong to a specific class. A False Positive (FP) occurs when the model wrongly assigns a sample to the wrong class. In contrast, a False Negative (FN) occurs when the model assigns the wrong class to a sample, even if the sample truly belongs to the proper class.

E. Phase 5: Hyperparameter Tuning

The hyperparameter adjustment achieved the optimal model's performance. Two hyperparameters, learning rate and batch size, were examined in this research. According to [36], the learning rate is a hyperparameter that controls how much a deep learning model changes whenever the model's weights are changed in response to the estimated error. Batch size, on the other hand, is the quantity of samples processed before updating the model [37]. According to several research [38], [39], learning rate and batch size significantly affect a neural network's performance.

Learning rates of 0.01, 0.001 and 0.0001 were tested during hyperparameter tuning. As for batch size, batch sizes of 32 and 64 were tested. A full grid search was performed during the hyperparameter tuning process, and thus, six experiments were performed for each image classification model.

The hyperparameter tuning process is realized using the Keras Tuner library in Python, and the models were allowed to be trained for 50 epochs. After hyperparameter tuning was performed for each image classification model, the model with the best validation accuracy was rebuilt, and the model was then tested with the test dataset to obtain the test accuracy.

Subsequently, the performance of the best-performing model for each image classification model, whether it is the custom CNN model proposed here or the transfer learning model, will be evaluated, compared and discussed in a subsequent section.

4. Results and Discussion

Extending upon the aforementioned technique, we carried out a sequence of tests to evaluate the resilience and precision of our model. The next part presents a comprehensive examination of the experimental configuration and the obtained results; we will Split this chapter to two section results and discussion.

A. Results

We will show all results that we achieved in our study

1) Training and Validation Loss

Figure 9 shows the training and validation loss of the model during the hyperparameter tuning process. In general, the training and validation loss reduces as the epoch increases. However, some trials failed to converge and remain stationary across epochs, especially for trial 1 and trial 2 of transfer learning models, where the learning rate for both was 0.01. While trial 5 and trial 6, which use a learning rate of 0.0001, show the lowest training and validation loss during the model training of transfer learning models, both trials had the highest loss during the training of the proposed CNN networks.

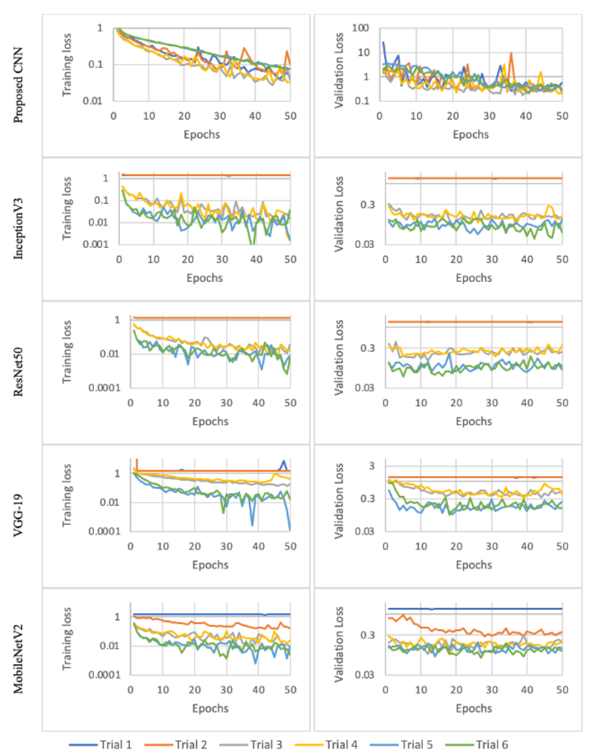

Figure 9. Training and validation loss of models plotted on log scale during the hyperparameter tuning process.

2) Validation Accuracy

Table IV summarizes the proposed CNN and transfer learning models' best validation accuracy for different learning rates and batch sizes during the hyperparameter tuning process. In general, in this study, a larger batch size of 64 would result in better validation accuracy than a smaller batch size of 32; as for the effects of learning rate, smaller learning rates result in better validation accuracy for transfer learning models. However, this trend is not observed in the proposed CNN model, where the highest validation accuracy is observed when the learning rate is 0.001 instead.

It is worth noting that for all the transfer learning models trained with a learning rate of 0.01, except for the MobileNetV2 model trained with a batch size of 64, the accuracies of the models were very low at less than 40%.

TABLE IV. The Validation Accuracy With diffrent Learning Rate and Batch Size

Trial	Learning Rate	Batch Size	Proposed CNN	InceptionV3	ResNet50	VGG-19	MobileNetV2
1	0.01	32	92.7 %	25.2%	25.4%	25.6%	25.9%
2	0.01	64	93.9%	25.4%	32%	26%	91.5%
3	0.001	32	95.1%	97.4%	95.9%%	90%	97.3%
4	0.001	64	94.6%	97.9%	96%	89%	97.2%
5	0.0001	32	91.7%	98.7%	98.5%	97.7%	97.3%
6	0.0001	64	92.1%	98.9%	98.6%	97.9%	98.3%

This was due to the models failing to converge during model training, as evident in the training and validation loss curves of those models shown in Figure 8, which do not decrease as the training epochs increase.

3) Optimal Hyperparameters

The highest validation accuracies for each model in the hyperparameter tuning process were bolded in Table V. Accuracy values, together with their corresponding hyperparameters, were summarized in Table V. From the hyperparameter tuning process, the best-performing proposed CNN model, which had a learning rate of 0.001 and batch size of 32, had a validation accuracy of 95.1%. As for other transfer learning models, the highest validation accuracies were attained with a learning rate 0.0001 and batch size of 64. The validation accuracies were 98.9%, 98.6%, 97.9% and 98.3% for InceptionV3, ResNet50, VGG- 19 and MobileNetV2, respectively. Based on this validation accuracy score, InceptionV3 provided the best validation accuracy, followed by ResNet50, MobileNetV2, VGG-19 and the proposed CNN network.

In addition, the best-performing models were tested with the test dataset to obtain the test accuracy. In general, the test accuracy was comparable to the validation accuracy. However, the sequence in terms of the test accuracy performance is different. Both ResNet50 and VGG-19 had a test accuracy of 98.7%, followed by MobileNetV2 with 98% test accuracy and InceptionV3 with 97.7%. The proposed CNN model had the lowest test accuracy compared to the other models at 94.4%.

B. Discussion

The reason the proposed CNN model could perform better than other transfer learning models could be attributed to insufficient model training. As seen from the CNN model's loss curves, stationarity had yet to be obtained by the end of the 50 epochs. This differs from the transfer learning models, where stationarity is observed towards the end of the 50 epochs. Thus, the proposed CNN model had not converged, resulting in lower accuracy. In addition, the transfer learning models used here were created by industry experts in CNN and initialized with pre-trained weights, which have been optimized with the training of the ImageNet dataset. Thus, convergence can be attained earlier with the transfer learning models trained with the new satellite images. The proposed CNN model can be trained with more epochs until convergence is attained, and hyperparameter tuning with more hyperparameters can be performed as part of future work to attain better performance with the model.

InceptionV3 attains the maximum validation accuracy compared to all other models, indicating its robust capacity to generalize well to unfamiliar data. The design of the building, which encompasses components of various sizes, enhances its exceptional performance.

MobileNetV2 has a robust test accuracy that is equivalent to that of ResNet50. Due to its lightweight design, this technology is well-suited for mobile applications while maintaining a high performance level.

Achieving an accuracy of 98.7%, ResNet50 and VGG-19 models demonstrate remarkable performance on the provided dataset. The high level of accuracy indicates that the models successfully capture the fundamental patterns and characteristics in the data, resulting in accurate predictions for most cases. This suggests that these models are highly efficient for the given task, making them suitable for real-world applications where precision is crucial.

5. Conclusions and Future Work

This study introduces a novel approach to road feature classification using convolutional neural networks (CNNs) and transfer learning. For this task, we contrast the accuracy of the suggested CNN model with that of other transfer learning algorithms, including ResNet50, MobileNetV2, VGG19, and InceptionV3. Various hyperparameters, including learning rate and batch size, are investigated in this study. The results demonstrated that the suggested CNN and the TL models, which had undergone hyperparameter tuning, can classify roundabouts, crosswalks, intersections, and overpasses with relatively high accuracy. The best-performing model during the validation phase is InceptionV3, with an accuracy of 98.9%, whereas the best-performing model during the test phase is the ResNet50 and VGG-19 models, with an accuracy of 98.7%. The proposed CNN model got 95.1% and 94.4% accuracy during the validation and test stage.

The implications of our research extend to various domains within intelligent transportation systems, including autonomous driving, traffic management, and road infrastructure maintenance. Accurate classification of road features enables safer navigation, improved traffic flow, and adequate transportation planning and management decision-making.

Several obstacles were faced throughout the study, such

TABLE V. Validation accuracy and Test Accuracy of Image Classification Models with their Most Optimal Hyperparameters

Model	Hyperparameters	Validation Accuracy (%)	Test Accuracy (%)
Proposed CNN	Learning rate: 0.001, Batch size: 32	95.1%	94.4%
InceptionV3	Learning rate: 0.0001, Batch size: 64	98.9%	97.7%
ResNet50	Learning rate: 0.0001, Batch size: 64	98.6%	98.7%
VGG-19	Learning rate: 0.0001, Batch size: 64	97.9%	98.7%
MobileNetV2	Learning rate: 0.0001, Batch size: 64	98.3%	98%

as the need for thorough hyperparameter tweaking to get the most favourable model performance. This issue was resolved by conducting systematic experiments with different learning rates and batch sizes, enhancing accuracy. Moreover, the constrained magnitude of the dataset presented difficulty in efficiently training the suggested CNN model. In order to address this issue, data augmentation methods were used to artificially increase the size of the dataset, hence improving the model's capacity to make accurate predictions on new and unknown data.

Future research may include supplementary road feature categories, such as traffic signs and signals, to establish a more all-encompassing classification system. Further investigations can also focus on optimizing the model architecture, refining transfer learning strategies, and incorporating contextual information for more comprehensive road analysis.

References

[1] K. Jurczenia and J. Rak, "A survey of vehicular network systems for road traffic management," *IEEE Access*, vol. 10, pp. 42 365–42 385, 2022.

[2] N. Sakib, T. Paul, M. T. Ahmed, K. Al Momin, and S. Barua, "Investigating factors influencing pedestrian crosswalk usage behavior in dhaka city using supervised machine learning techniques," *Multimodal Transportation*, vol. 3, no. 1, p. 100108, 2024.

[3] A. Sarker, H. Shen, M. Rahman, M. Chowdhury, K. Dey, F. Li, Y. Wang, and H. S. Narman, "A review of sensing and communication, human factors, and controller aspects for information-aware connected and automated vehicles," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 1, pp. 7–29, 2019.

[4] M. M. Antony and R. Whenish, "Advanced driver assistance systems (adas)," in *Automotive Embedded Systems: Key Technologies, Innovations, and Applications*. Springer, 2021, pp. 165–181.

[5] M. Rizzo, M. Marcuzzo, A. Zangari, A. Gasparetto, and A. Albarelli, "Fruit ripeness classification: A survey," *Artificial Intelligence in Agriculture*, vol. 7, pp. 44–57, 2023.

[6] I. H. Sarker, "Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions," *SN computer science*, vol. 2, no. 6, p. 420, 2021.

[7] E. Ranyal, A. Sadhu, and K. Jain, "Road condition monitoring using smart sensing and artificial intelligence: A review," *Sensors*, vol. 22, no. 8, p. 3044, 2022.

[8] D. K. Dewangan and S. P. Sahu, "Rcnet: road classification convolutional neural networks for intelligent vehicle system," *Intelligent Service Robotics*, vol. 14, no. 2, pp. 199–214, 2021.

[9] V. Tümen and B. Ergen, "Intersections and crosswalk detection using deep learning and image processing techniques," *Physica A: Statistical Mechanics and its Applications*, vol. 543, p. 123510, 2020.

[10] H. Li, M. Hu, and Y. Huang, "Automatic identification of overpass structures: A method of deep learning," *ISPRS Int J Geoinf*, vol. 8, no. 9, p. 421, 2019.

[11] U. Jilani, M. Asif, M. Rashid, A. A. Siddique, S. M. U. Talha, and M. Aamir, "Traffic congestion classification using gan-based synthetic data augmentation and a novel 5-layer convolutional neural network model," *Electronics (Basel)*, vol. 11, no. 15, p. 2290, 2022.

[12] K. S. Sandeep *et al.*, "Road feature extraction using cnn and gru in edge-enabled vehicular cloud networks," in *2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*. IEEE, 2024.

[13] S. Mekruksavanich *et al.*, "Deep learning approaches for classification of road types using wearable sensors embedded in smart glasses," in *2023 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT NCON)*. IEEE, 2023.

[14] K. Liao, "Road damage intelligent detection with deep learning techniques," in *2022 IEEE 5th International Conference on Information Systems and Computer Aided Education (ICISCAE)*. IEEE, 2022.

[15] K. Bayoudh, F. Hamdaoui, and A. Mtibaa, "Transfer learning based hybrid 2d-3d cnn for traffic sign recognition and semantic road detection applied in advanced driver assistance systems," *Applied Intelligence*, vol. 51, no. 1, pp. 124–142, 2021.

[16] A. A. Alhussan *et al.*, "Pothole and plain road classification using adaptive mutation dipper throated optimization and transfer learning for self driving cars," *IEEE Access*, vol. 10, pp. 84 188–84 211, 2022.

[17] M. Rezapour and K. Ksaibati, "Convolutional neural network for roadside barriers detection: Transfer learning versus non-transfer learning," *Signals*, vol. 2, no. 1, pp. 72–86, 2021.

[18] S. H. Hosseini *et al.*, "Road sign classification using transfer learning and pre-trained cnn models," in *International Conference on Artificial Intelligence and Smart Vehicles*. Springer Nature Switzerland, 2023.

[19] N. Saeed, R. G. Nyberg, and M. Alam, "Gravel road classification based on loose gravel using transfer learning," *International Journal of Pavement Engineering*, vol. 24, no. 2, p. 2138879, 2023.

[20] K. K. M. Shariff *et al.*, "Wet road detection using cnn with transfer learning," in *2022 IEEE 12th Symposium on Computer Applications Industrial Electronics (ISCAIE)*. IEEE, 2022.

[21] X. Qi, P. Zhu, Y. Wang, L. Zhang, J. Peng, M. Wu, J. Chen, X. Zhao, N. Zang, and P. T. Mathiopoulos, "Mlrsnet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 169, pp. 337–350, 2020.

[22] B. Patel, T. D. Barfoot, and A. P. Schoellig, "Visual localization with google earth images for robust global pose estimation of uavs," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2403–2409.

[23] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, Oct 2018.

[24] D. Dablain, K. N. Jacobson, C. Bellinger, M. Roberts, and N. V. Chawla, "Understanding cnn fragility when learning with imbalanced data," *Machine Learning*, vol. 113, no. 7, pp. 4785–4810, 2024.

[25] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J Big Data*, vol. 6, no. 1, p. 27, Dec 2019.

[26] S. Connor and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[27] K. Potdar, T. S. Pardawala, and C. D. Pai, "A comparative study of categorical variable encoding techniques for neural network classifiers," *International journal of computer applications*, vol. 175, no. 4, pp. 7–9, 2017.

[28] M. Mohiuddin, M. S. Islam, and M. H. Kabir, "Performance analysis of bearing fault diagnosis using convolutional neural network," in *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*. IEEE, 2021.

[29] V. E. Balas, R. Kumar, R. Srivastava *et al.*, *Recent trends and advances in artificial intelligence and internet of things*. Springer, 2020.

[30] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 12, pp. 6999–7019, 2021.

[31] J. Gupta, S. Pathak, and G. Kumar, "Deep learning (cnn) and transfer learning: a review," in *Journal of Physics: Conference Series*, vol. 2273, no. 1. IOP Publishing, 2022, p. 012029.

[32] S. Mascarenhas and M. Agarwal, "A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification," in *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*, vol. 1. IEEE, 2021, pp. 96–99.

[33] Y. Gulzar, "Fruit image classification model based on mobilenetv2 with deep transfer learning technique," *Sustainability*, vol. 15, no. 3, p. 1906, 2023.

[34] G. Meena, K. K. Mohbey, A. Indian, and S. Kumar, "Sentiment analysis from images using vgg19 based transfer learning approach," *Procedia Computer Science*, vol. 204, pp. 411–418, 2022.

[35] A. Nivrito and M. R. B. Wahed, "Comparative analysis between inception-v3 and other learning systems using facial expressions detection," Ph.D. dissertation, BRAC University, 2016.

[36] S. Basodi, C. Ji, H. Zhang, and Y. Pan, "Gradient amplification: An efficient way to train deep neural networks," *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 196–207, 2020.

[37] J. Brownlee, "What is the difference between a batch and an epoch in a neural network," *Machine learning mastery*, vol. 20, pp. 1–5, 2018.

[38] G. Bruno, D. Antonelli, and D. Stadnicka, "Evaluating the effect of learning rate, batch size and assignment strategies on the production performance," *Journal of Industrial and Production Engineering*, vol. 38, no. 2, pp. 137–147, 2021.

[39] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT express*, vol. 6, no. 4, pp. 312–315, 2020.