

# Optimization for Pre-trained CNNs

Marzena Halama<sup>α</sup>

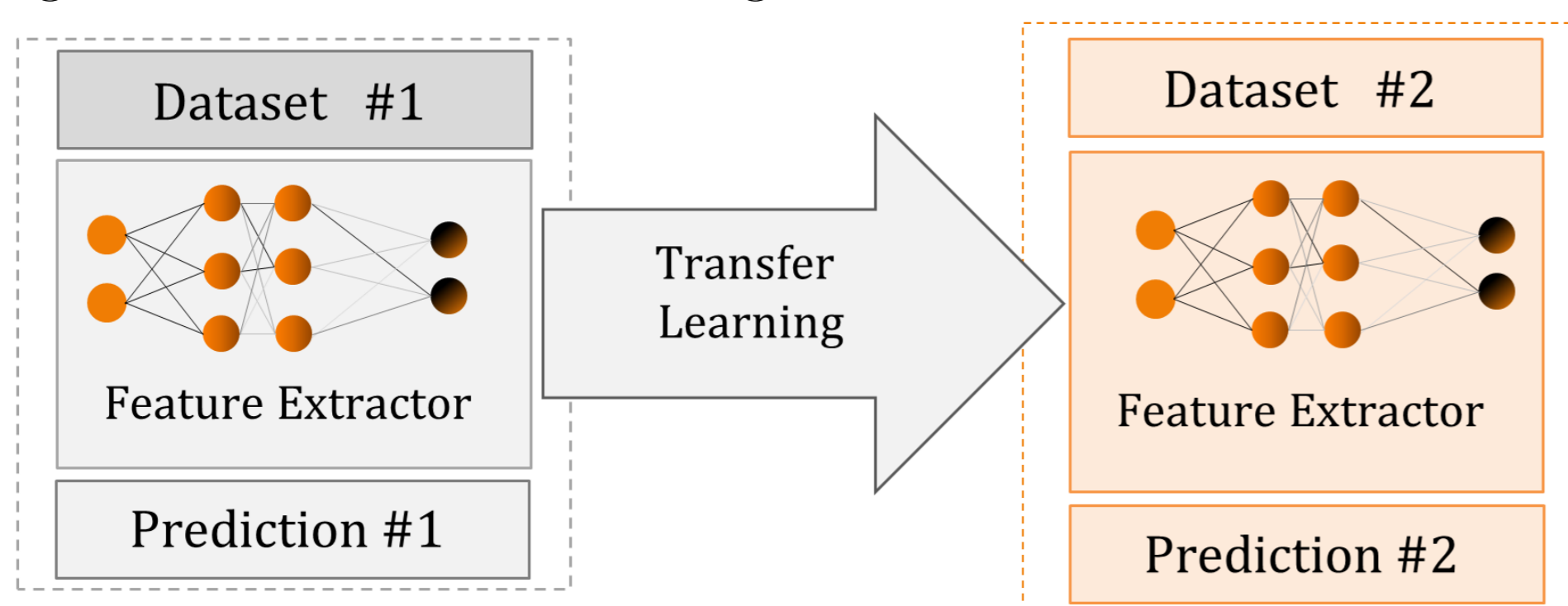
<sup>α</sup>Institute of Theoretical and Applied Informatics, Polish Academy of Science, Gliwice, Poland



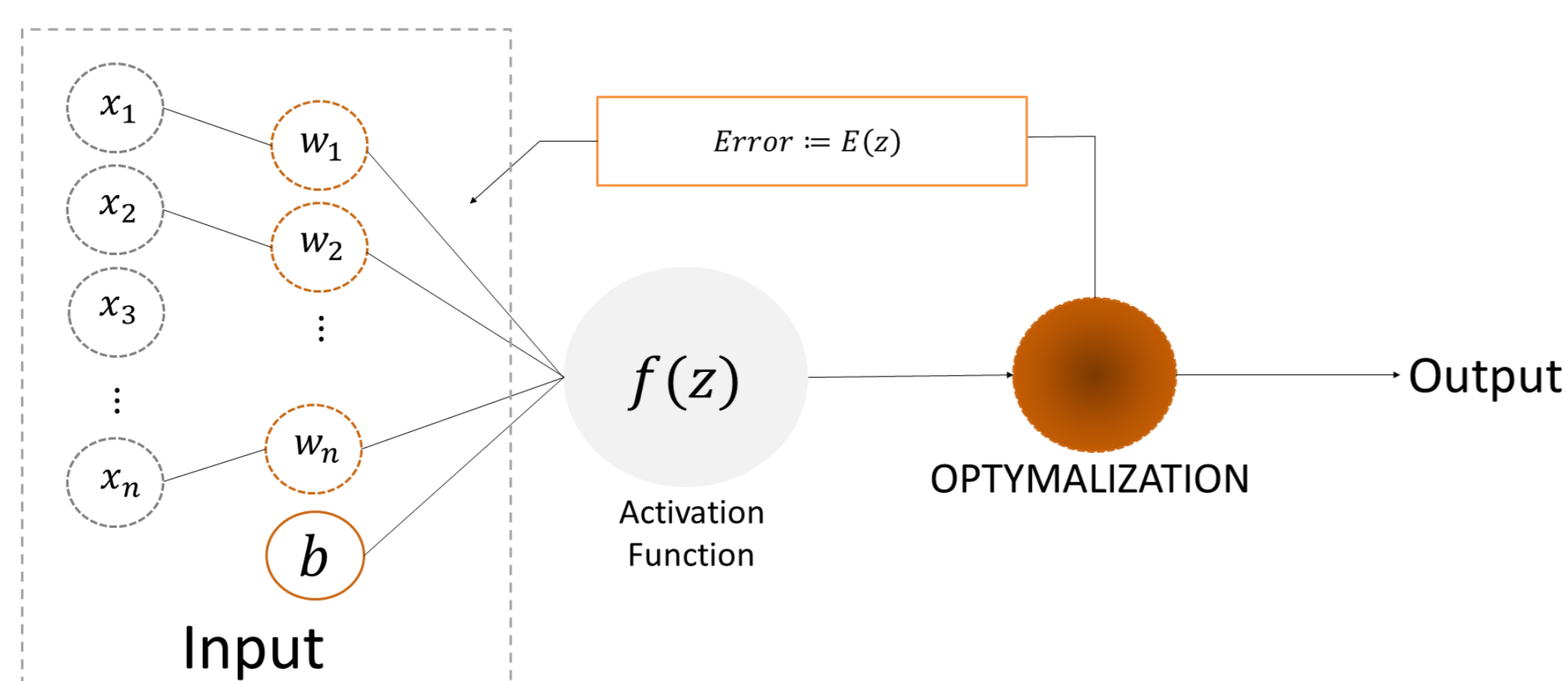
Convolutional neural networks (CNNs) constitute a fundamental cornerstone of computer vision. With increasing complexity, the need for effective optimisation strategies remains crucial. Techniques such as transfer learning (TL), utilising pre-trained networks, enable the deployment of advanced models on mobile devices with limited computing capacity, including autonomous vehicles and educational applications. The study explores optimisation strategies for Keras models, focusing on the impact of different algorithms on performance and accuracy. The results demonstrate that appropriate optimiser selection enhances learning efficiency, mitigates overlearning, and supports accurate image recognition.

## BACKGROUND

CNNs, recognized for their feature extraction capabilities, are widely applied in computer vision, particularly with large datasets such as ImageNet. Due to the resource-intensive nature of many CNN architectures, TL is increasingly being used to reduce the need for additional training and support the development of mobile applications for resource-constrained devices. The diagram below illustrates the general workflow of TL.



The selection of an optimisation algorithm is crucial for tuning pre-trained CNNs. This choice impacts the speed, robustness, and generalisability of the model, which is particularly significant in TL. The image below represents the optimisation process in deep learning. This process facilitates the deployment of state-of-the-art models in mobile applications with limited computational resources.



## METHODOLOGY

**Transfer learning** utilises pre-trained models that act as feature extractors with previously trained weights. These weights are adapted to new tasks, enabling efficient use of CNNs in mobile applications with minimal computational effort. **Pre-trained CNN models** Popular mobile-friendly models available in Keras were analysed. MobileNetV2 employs depth-separable convolutions, offering efficiency for mobile applications [2]. EfficientNetV2B0 provides scalability, balancing speed and accuracy through compact scaling [3]. Xception leverages deeply separable convolutions, making it lightweight and effective [1]. ResNet50V2 incorporates residual connections, facilitating the training of deep networks [4]. VGG16 relies on simplicity, using small 3x3 filters and a uniform layer structure [4]. **Optimization Methods for CNNs** Several optimization methods were used to assess their impact on training performance and accuracy, including gradient descent (GD), stochastic gradient descent (SGD), momentum, Adagrad, RMSprop, and Adam. **Datasets** Two datasets were considered: the generic ImageNette database and a custom ToyVision set comprising images of objects not present in ImageNet. The focus was on object categories, excluding non-object categories, to tailor the solution for mobile applications. The results are applicable beyond EduToyz and ImageNette datasets, extending to various data types and environments.

Dataset	Training Set	Validation Set	Test Set
ImageNette	24	24	2648
ToyVision	24	24	552

The table above shows the distribution of data sets in different subsets (training, validation and test sets) for two data sets: ImageNette and ToyVision. **Evaluation** Average loss function values were compared for each dataset using the selected optimization methods.

## RESULTS

This work presents a approach for creating object classifiers for mobile devices at low computational resource cost. The results of this study highlighted the key role of the optimisation algorithm for multi-view image classification. The choice of appropriate optimisation methods significantly affects on the performance of pre-trained CNNs and leads to improved predictive accuracy of machine learning models. Five pre-trained models from the keras collection were used. Their overall performance in the task was tested for two databases containing two different datasets: ImageNette and ToyVision.

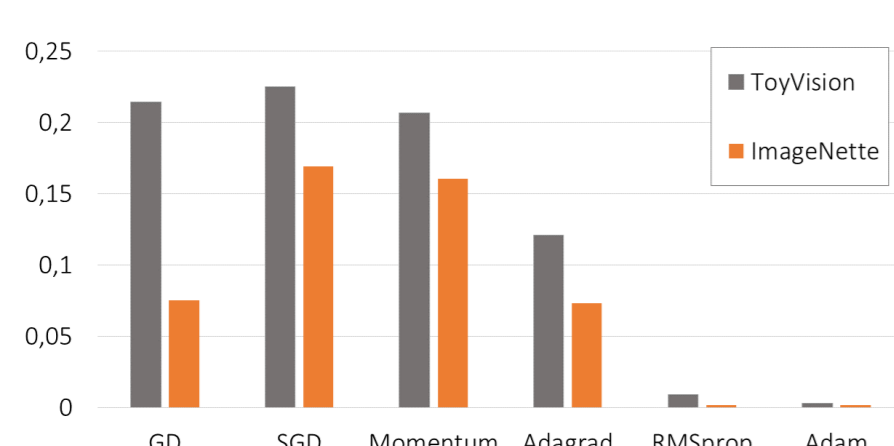


Figure 1: MobileNetV2

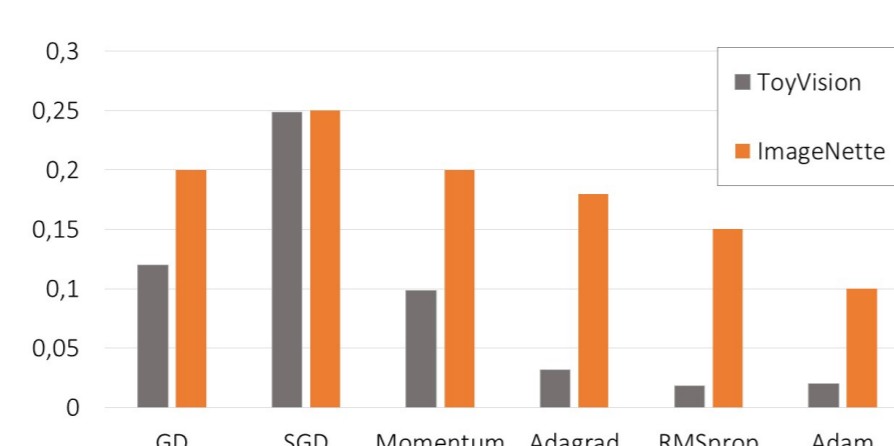


Figure 2: EfficientNetV2B0

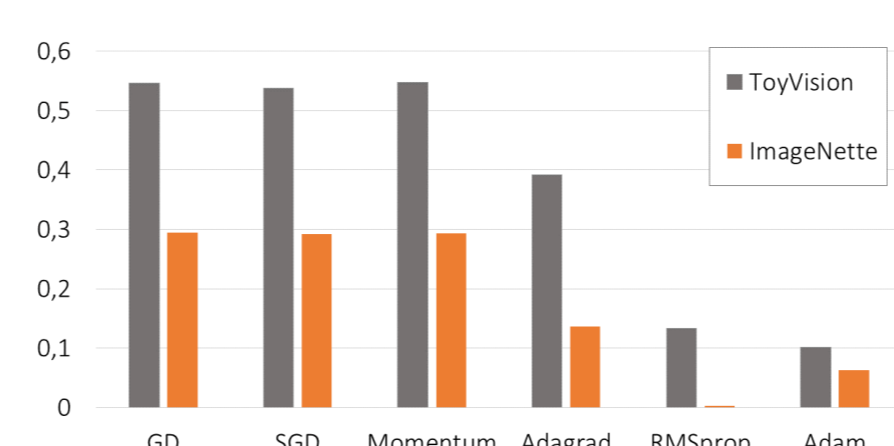


Figure 3: Xception

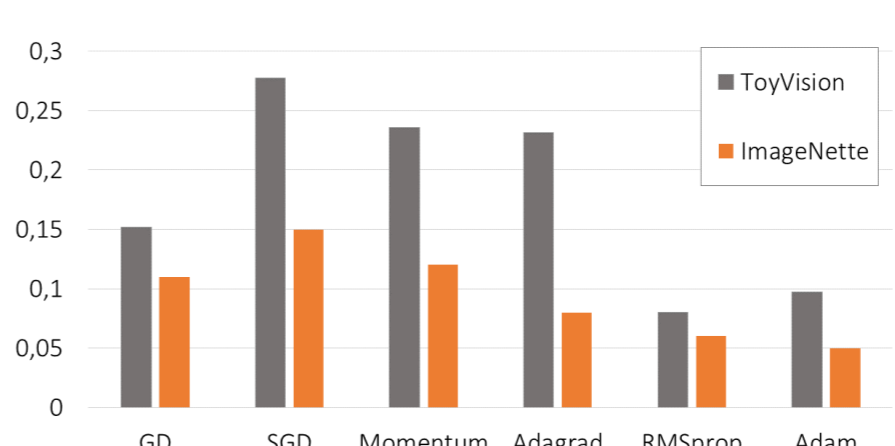


Figure 4: ResNet50V2

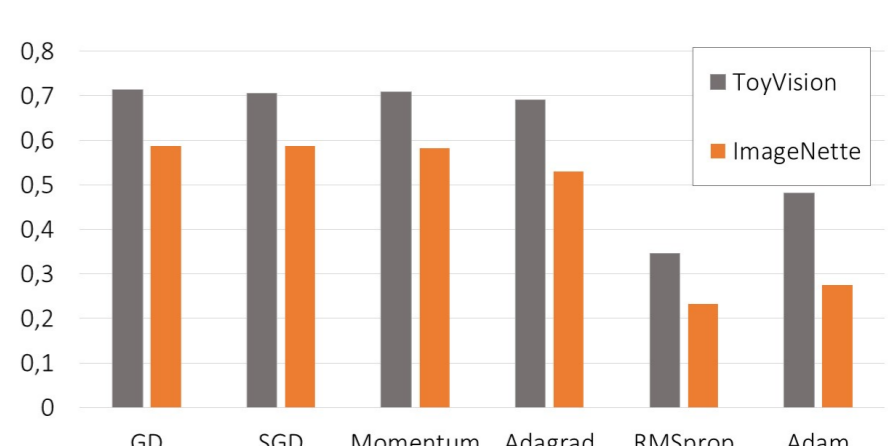


Figure 5: VGG16

In summary, the analysis highlights that RMSprop consistently outperforms other optimisation algorithms, achieving the lowest loss values across all neural network architectures tested on the ImageNette dataset. While Adam is often considered a versatile optimiser, its performance is not sufficient for some models, such as MobileNetV2 and EfficientNetV2B0, in this particular context. These results suggest that RMSprop and Adam are the most effective optimisation options, depending on the model and dataset used.

### References

- [1] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
- [2] Debjyoti Sinha and Mohamed El-Sharkawy. Thin mobilenet: An enhanced mobilenet architecture. In *2019 IEEE UEMCON*. IEEE, 2019.
- [3] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, 2019.
- [4] Sasha Targ, Diogo Almeida, and Kevin Lyman. Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*, 2016.

