1 Abstract

Today Machine Learning models and Computer Vision Applications are available for a variety of practical tasks. As part of Ambient Assisted Living(AAL) it should be evaluated whether to what extent they are useful for assistive Applications. The very well-known card game called “Dobble” should serve as a use case. The Dobble deck consists of about 55 cards, each card with 8 simple symbols. The symbols are selected from a group of 57 symbols in such a way that exactly one symbol matches on any two cards. In this project, A Machine Learning Application is developed where the user plays a card game with machine. Several Convolutional Neural Networks(CNN) are examined for suitability and per- formance with regard to Image Labelling and Object Detection. For Object Detection YOLO ver- sion 5 (You Only Look Once) is implemented and tested for real time symbol detections. Finding of better neural architecture either through Image Classification or Object Detection is evaluated. For this purpose classical Neural networks are first trained with frameworks like Pytorch and Ten- sorFlow and then Machine Learning based Application of Dobble is implemented. The training, testing accuracies and losses are examined to select the better neural architecture. Even though YOLOv5 is failed to detect real time detections of 16 symbols per frame, it is very highly difficult to detect multiple symbols in real time frames through web cam. The techniques like Transfer Learning called Fine Tuning and Feature Extraction have been applied on Classical Neural Network Architectures like ResNet, InceptionNet, DenseNet etc.. to train with our own dobble card image dataset to check whether the network is able to learn or classify 55 classes. But the networks face difficulty to extract HOG(Histogram of Gradients) fea- tures on card images in order to learn 8 symbols on each card and to classify 55 images because every card has one common symbol. The Language used is Python Object Oriented Programming(OOP) to interfere with these Frameworks Pytorch and TensorFlow. These Networks are trained on environments like Google Colab Pro for cloud training and Jupyter Notebook to train in a local computer. To deploy the trained weights on a Computer Vision Application is OpenCV to make real time detections. The persons who were facing problems with less Perception capabilities and Facial recognition senses, mostly elderly persons who stays at home or leading independent life can train their visual recep- tors by playing the game with machine. Thus the project has been able to accomplish the Machine Learning based Application to create a competition between user and machine.

Contents

List of Figures

2 INTRODUCTION

2.1 AMBIENT ASSISTED LIVING

2.2 MACHINE LEARNING INFLUENCE ON AMBIENT ASSISTED LIVING

2.3 CONVOLUTIONAL NEURAL NETWORKS

2.4 CROSS ENTROPY LOSS FUNCTION

hhjdjdjkdskksdkskds

2.5 ACTIVATION FUNCTION IN NEURAL NETWORKS: ReLU (RECTIFIED LINEAR UNIT)

2.6 OVERFITTING

2.7 DROPOUT REGULARIZATION TECHNIQUE

2.8 BATCH NORMALIZATION

2.8.1 OPTIMIZATION

Stochastic Gradient Descent

Adam Optimization (Adaptive Moment Estimation)

3 Material and methods

3.1 Pretrained networks

Pre-trained networks are models that have been previously trained for another task or large datasets and can be fine-tuned for specific purpose [2]. They can significantly reduce the training time and improve the performance of a model, especially when there is limited data [2]. Pretrained network that used for this study are

- ResNet50: The ResNet50 [3]is architected to act as a bottleneck, improving the training process. It employs residual mapping to address the issue of saturation degradation. Pre-trained ResNet50 models are indeed capable of capturing complex facial features [4] .
- GoogleNet (Inception): GoogleNet [9], also known as Inception, is a deep convolutional neural network architecture. It uses inception modules to efficiently use computational resources while maintaining a large receptive field. This can be beneficial for capturing different scales of facial expressions [1].
- AlexNet: AlexNet [7] is one of the pioneer deep learning network. It has five convolution layers followed by three fully connected layers.It incorporates convolutional layers, max pooling layers, and dense layers as its fundamental components [6].

3.2 Transfer learning

This method is highly practical and efficient for tasks with limited data availability. The process of transfer learning involved training the convolution neural network model using a substantial amount of data. subsequently, the model undergoes a fine-tuning phase to train on a smaller, specific dataset [1].

3.2.1 Fine-tuning

Fine-tuning is a process that takes a pre-trained model and “tunes” it for a different but related problem. This uses weights from a pre-trained model as the initialization for a new model [8]. fine-turning steps involved are

1. Pretraining: pretrained network such as GoogleNet [9], AlexNet [7] and resNet50 [3] was used. This allows the model to learn general features from a wide variety of images.
2. Creating the model: A new neural network model is created by using the architecture and parameters of the pretrained Networks, except for the output layer [5].
3. Adding the Output Layer: The target model is enhanced with an output layer. The total number of outputs is determined as seven, which corresponds to the total number of classes in the target dataset.

4. Training the Model: The target model is then trained on the target dataset. The output layer's parameters are newly learned, while the other layers' parameters are fine-tuned based on the parameters of the pretrained network [5].

3.2.2 Feature extraction

References

[1] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8:1–74, 2021.

[2] Baeldung. Neural network pre-training. `https://www.baeldung.com/cs/neural-network-pre-training`.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Shantala S Hiremath, Jayaprada Hiremath, Vaishnavi V Kulkarni, BC Harshit, Sujith Kumar, and Mrutyunjaya S Hiremath. Facial expression recognition using transfer learning with resnet50. In *Inventive Systems and Control: Proceedings of ICISC 2023*, pages 281–300. Springer, 2023.

[5] Dive into Deep Learning. Fine-tuning, 2024.

[6] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *CoRR*, abs/1404.5997, 2014.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[8] Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. *arXiv preprint arXiv:2305.16938*, 2023.

[9] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.