Journal of Information Systems Engineering and Management
2025, 10(31s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Pocket-Sized AI: Evaluating Lightweight CNNs for Real-Time Sketch Detection

Nilkamal More[1], Suchitra Patil[1], Abhijeet Pasi[1], Bhakti Palkar[1], V.Venkatramanan[1], Pankaj Mishra[1]
K.J.Somaiya School of engineering, *Somaiya Vidyavihar University, Vidyavihar, Mumbai, India*

ARTICLE INFO

Received: 26 Dec 2024

Revised: 14 Feb 2025

Accepted: 22 Feb 2025

ABSTRACT

Portability with Performance Balance of neural network model is of utmost importance in the current era of Mobility. Applications and uses of deep learning neural network are growing at a fast pace and it is necessary for neural network model to be portable but capable enough to learn and identify tasks on the computational power of mobile devices. In this paper, the performance and practicability of regular Convolutional Neural Network (CNN) & depthwise separable CNN (MobileNet) are discussed and compared on mobile phones by carrying out some parametric modifications and trials on resources to study accuracy tradeoffs. Subsequently, we illustrated the implementation of depthwise separable CNN (MobileNet) on a sketch-based recognition game rewarding for successful recognition of sketch which is provided as an assignment on android platform.

Keywords: Depth-wise separable convolution, MobileNet, sketch recognition, neural network, convolution.

INTRODUCTION

Convolutional Neural Networks (CNNs) are the go-to method in computer vision problems, particularly since AlexNet won the 2012 ImageNet Competition, which really saw a lot of interest come their way for deep convolutional architectures. Ever since, the general trend has been going deeper and more complex neural architectures to deliver higher accuracy and lesser prediction errors. Though the outstanding accuracy provided by CNNs is undeniable, they are endowed with some huge drawbacks. High-performing CNN models like AlexNet contain billions of parameters, which when retained in floating-point precision require a lot of memory. In addition, existing CNN models have continued to grow in depth, necessitated by the size of larger training sets and the need for higher accuracy—e.g., residual networks (ResNets) tend to have hundreds or even thousands of layers. Timely and effective computation has thus become extremely critical, particularly with the limitations faced by mobile computing environments.

A number of practical issues highlight the necessity for smaller, more efficient neural network structures. First, most real-world use cases, including image classification or retrieval on mobile phones, demand CNNs to operate on the device itself. Running big models, taking tens or hundreds of megabytes of storage, is generally impossible on mobile phones. Second, it is now impossible to train very deep neural networks on a single piece of computing hardware because of limits in memory and computational capacity. Finally, low-power hardware deployments of deep neural networks necessarily require more efficient, lighter-weight network architectures.

In order to solve these problems, the depthwise separable convolutional neural network model, the prime example of which is MobileNet, was proposed. MobileNet is a leaner version of standard CNN architectures that significantly reduces the computational cost and results in a small neural model ideal for mobile or low-resource devices with hardly any accuracy compromise.

The rest of this paper is organized as follows: Section 2 outlines related literature and existing work on neural network architectures. Section 3 outlines specifications of the neural network models considered in this research. Section 4 is where experimental results and comparisons of the two architectures are given. Section 5 offers insights into a sketch-based recognition game employed in testing, and lastly, Section 6 presents key findings and conclusions.

OBJECTIVES

The system needs to identify what the user intends to draw through their freehand sketch. The main objective involves teaching machines to recognize abstract human ideas expressed through sketches just as humans do when interpreting such visuals.

The system utilizes this understanding to develop interactive applications such as sketch-based games which require real-time interpretation of sketches to support gameplay that depends on abstract representation recognition.

The game basically is a set of this sketch tasks that a user needs to be completed in time and the neural network tries to guess the sketch. If it finds it to be true, user is rewarded with points based on what & how the sketch is drawn.

LITERATURE SURVEY

The researcher discusses sketch classification with Convolutional Neural Networks (CNNs) in [1] which remains an understudied field in its area of study. Traditional techniques based on edge detection and feature extraction and machine learning algorithms including SVMs and random forests were used extensively in previous research. The paper introduces a new deep CNN structure which is designed to classify sketches. The proposed architecture includes convolutional layers to extract features from the sketch and a pooling layer to reduce data dimensions and a fully connected layer for classification purposes. A massive dataset of labeled hand-drawn sketches is used for training which applies backpropagation to minimize a loss function that evaluates classification accuracy. The implemented methodology outperforms current sketch classification approaches when tested on the benchmark TU-Berlin dataset thus validating CNNs as a promising direction for future research.

In this paper, the authors discuss the particular issues of sketch recognition, which is considered as a domain with a lot of diversity in hand-drawn sketches. They come up with a new deep CNNs architecture that is tailored for sketch processing. The framework includes a modified ResNet architecture that has incorporated residual connections and batch normalization to enhance the training process. Network adaptations involve reducing the sizes of the filters and the use of global average pooling in order to account for the low resolution of the sketches. Dropout layers have also been added in a manner that will ensure they do not cause overfitting.

The authors validate their approach with several different CNN architectures including AlexNet and VGG-16, and apply a large publicly available dataset of sketches to compare them. The author's rigorous methodology not only presents a new DCNN architecture, but also established a benchmark for future research in this topic and enable the exploration of other deep learning architectures for the recognition of sketches. The paper [5] transformed the trajectory of the computer vision by introducing deep learning and most importantly Convolutional Neural Networks (CNNs) to large-scale image classification on the ImageNet dataset.

The authors presented a deep convolutional neural network architecture which was subsequently known as AlexNet, consisting of five convolutional layers, max-pooling layers, and three fully connected layers. In order to solve the vanishing gradient problem, the authors[6] utilized rectified linear units (ReLUs) as activation functions, and for preventing overfitting, they incorporated dropout layers. The deep network was trained on the huge ImageNet dataset containing more than 1.2 million images to 1000 classes, which was computationally expensive, but the authors had access to fast GPU implementations and data augmentation. The proposed architecture performed well, with the top-1 error rate at 37.5% and the top-5 error rate at 17.0%, which outperformed all state-of-the-art approaches at the time. In research paper [10], a new deep learning framework for sketch classification has greatly contributed to the field by addressing problems such as inherent vagueness, low-quality data, and shallow learning.

The approach employs web images, learns from a large number of web images for each of the sketch categories, and features a dual stream architecture that extracts disparate types of information. It also involves multimodal fusion, wherein features of both streams are merged to depict the semantic concepts between images and sketches. SketchNet also includes a sketch-specific loss function that forces the model to focus on discriminative sketch-specific features [7]. TU-Berlin and Quick Draw datasets are employed in this study, and SketchNet consists of a sketch encoder, a web image encoder, a multimodal fusion layer, and a classifier. Results are shown that SketchNet has been surpassed by earlier work and it is possible to use external data. The research also demonstrates the importance of multimodal fusion and domain specific loss functions, which will find applications in future work on sketch understanding and applications such as sketch retrieval and sketch-based image generation. Other research directions are incorporating additional modalities to reach a better comprehensive understanding, researching more sophisticated deep learning architectures to reach higher precision and interpretability, and exploring domain adaptation methods for transferring SketchNet to novel sketch domains with little or no data. The research paper [8] introduced Sketch-a-Net: a deep neural network approach to address free-hand sketch recognition. It addresses the unique features of sketches, such as the multi-level of abstraction and sequentiality. The method has three distinctive features: a model structure and learning parameter selection for addressing the iconic and abstract nature of the sketches, a multi-channel architecture for representing the sequential ordering of strokes in a sketch, and a multi-scale network ensemble for addresssing variability in abstraction and sparsity. The article contrasts Sketch-a-Net with the best current alternatives and concludes that it outperforms all of them, even human beings, in sketch recognition. The system also proposes a multi-scale ensemble of networks to tackle abstraction level variability. In research paper [9] introduces a new solution for sketch classification, which normally is challenging due to its vagueness and lack of details relative to photographs.

The paper solves the limitations of conventional solutions, including shallow data, shallow learning, and intrinsic vagueness. Sketch-a-Net employs a new Deep Convolutional Neural Network (CNN) architecture, e.g., a two-stream architecture, web image pre-training, multimodal fusion, and sketch-specific loss function. The new model performs better than existing methods on two benchmark datasets, even beating human performance on the Quick Draw dataset. The paper also emphasizes the significance of web images, multimodal fusion, and sketch-specific loss functions in enhancing classification accuracy. Sketch-a-Net has significantly impacted the research community for sketch recognition by showing the power of pre-training on web images and deep learning in the classification of sketches, showing the efficacy of multimodal fusion and domain-adapted loss functions, and outlining the direction of future work on sketch understanding and applications such as retrieval and generation. Directions for the future are to include more modalities, investigate deeper architectures, and research domain adaptation methods for extending Sketch-a-Net to new sketch domains with limited data.

The research paper [11] presents a challenging task in computer vision where hand-drawn sketches are utilized for various tasks. Fei-Fei Li et al.'s 2005 article presents a novel method that makes use of local features and global shape cues. Local features are extracted using Gabor filters, and global shape cues are extracted using the moment of inertia shape descriptor. The local features and global shape cues are learned and classified using the support vector machine (SVM). The authors evaluated their method on 25,000 sketches from 250 object classes and obtained an accuracy of 66.7%, which is significantly higher than the previous methods. This paper is significant as it was the first to utilize both local features and global shape cues in the sketch-based object recognition task.

In research paper [7] have reported impressive results on image classification tasks, particularly using the AlexNet architecture. Nonetheless, conventional CNNs are hindered by vanishing gradients, computational expense, and inflexibility for various input sizes. One suggested approach is an Inception module, comprising convolutional filters of different sizes used in parallel on the same input. This method has enhanced gradient flow, reduced computational cost, and the capacity to capture multi-scale features. The network architecture employs stacked modules with increasing filter sizes and depth, global average pooling for dimensionality reduction, use of auxiliary classifiers for regularization, and normalization of activations for improved convergence. This architecture has been a starting point for state-of-the-art CNN architectures and inspired follow-up work on network design, hyperparameter search, and training algorithms.

Research paper [4] explain how Deep CNNs have been effective in computer vision applications but were too costly in terms of time and memory to be utilized in mobile and embedded systems. Deepwise separable convolutions achieve substantial parameter and operation reduction over regular convolutions. The efficient architecture design utilizes depthwise separable convolutions, linear bottleneck layers, average pooling, and global average pooling for classification. Two hyperparameters at the global level (α and β) control network width and depth, enabling easy model scaling under different resource budgets. The method achieved state-of-the-art ImageNet classification accuracy and demonstrated significant speed and memory footprint savings over popular architectures like VGG16 and Inception v3. MobileNets have made MobileNets a successful mobile and embedded vision architecture, which motivated further research into lightweight and efficient CNN designs. Future research directions involve investigating knowledge distillation, evolving to new hardware platforms, and examining applications in edge computing and augmented reality.

Paper [2] introduce their work on Convolutional Neural Networks (CNNs) and their worth for computer vision tasks in the paper (2015). Authors introduce CNN model results in image recognition through their analysis of AlexNet OverFeat GoogLeNet and He et al. CNNs are general-purpose feature extraction tools that the authors particularly study using feature maps. The authors show that feature maps learned by CNNs work better than initial CNN models when applied to classification tasks using suboptimal models. The authors demonstrate that using lower-layer features produces better results in classification tasks. The authors describe their upcoming research which involves studying different CNN architectures and datasets and investigating pre-trained models of AlexNet and GoogLeNet as well as implementing convolutional layers for classification and model replication of their work. This paper delivers a detailed analysis of CNN features together with their possible uses in computer vision applications.

This white paper written by [3] delivers extensive knowledge about Convolutional Neural Networks (CNNs) together with their applications in image recognition. This work examines CNN architecture together with its different layers as well as embedded system implementation issues and obstacles. The authors explore both specific implementations of traffic sign recognition and they discuss potential future applications of CNNs. The research investigates both the benefits and difficulties of applying CNNs to image recognition problems and general problems. The authors reference several neural network models and studies including "Face Recognition: A Convolutional Neural Network Approach" alongside "Deep Belief Networks" and "Long Short-Term Memory" and "Recurrent Neural Networks".

The paper [12] stresses that mobile sketch-based perception models need to be both lightweight and able to perform real-time. Standard deep learning models have shown high accuracy on well-structured datasets but their reliability in unstructured real-world environments is questionable. Depthwise separable convolutions (DSCs) are proposed as a potential way to enhance model efficiency, but the need for further research on compressing sketch recognition models and their effectiveness in mobile applications, including color, texture, and pressure, is strongly emphasized.

A study [13] presents an energy-efficient DSC accelerator designed for image recognition applications that reaches 413.2 GOPs throughput and 65.18 GOPs/W energy efficiency through the use of MobileNetV2 architecture. Although the work does not focus directly on sketch recognition, it shows that DSCs can greatly decrease network complexity and memory requirements, which is important for mobile devices with limited resources.

Another contribution [14] presents the Multi-Graph Transformer (MGT) which performs better than Inception V3 and MobileNetV2 in sketch recognition tasks and has faster inference times. The MGT model reaches an accuracy of 72.80% which is close to the CNN benchmark of 74.22% and benefits from the natural sparsity of sketch data. Its efficiency and responsiveness make it particularly suitable for real-time sketch recognition on mobile devices.

The evaluation of lightweight CNN models such as EfficientNet-B0 and MobileNet-V3 for mobile human action recognition demonstrates that these architectures provide performance that is equal to or superior to that of more complex models like ResNet-50. Although the results are not specifically applied to sketch recognition or DSCs, the results confirm the feasibility of deploying compact models for mobile inference tasks, which suggests the potential applicability to sketch-based systems as well [15]

The paper [16] introduces a deep convolutional neural network (DCNN)-based framework for hand-drawn sketch recognition, based on pre-trained models like VGGNet, ResNet, and Inception-v3. But it does not explore the performance and applicability of lightweight CNN architectures suitable for real-time sketch detection in miniaturized mobile AI setups. Experimental results demonstrate that the DCNN approach proposed in this paper can effectively surpass other state-of-the-art methods in both sketch classification and retrieval tasks, which demonstrates its effectiveness in handling high-level sketch data.

Studies[18 19] investigates a fine-tuned convolutional neural network (CNN) for sketch-based image retrieval with a focus on the application of deep neural representations for partially colored sketches. The proposed approach employs a good CNN structure, with training on a sketch-augmented dataset, for learning discriminative neural codes to enhance retrieval precision. The findings show that this approach is superior to other state-of-the-art methods in large-scale sketch-based image retrieval, especially on mobile platforms. While the emphasis is not necessarily on lightweight CNNs for real-time detection, the results present useful insights that can be used to guide the development and assessment of efficient models for mobile applications.

Study [20] describes Random Sketch Learning (Rosler) is an architecture designed to facilitate computationally effective small artificial intelligence (AI) for edge computing systems. It presents a compressing-while-training framework universally applicable, whereby models learn succinct representations directly while training, obviating the necessity for explicit computationally heavy pre-training or post-training compression steps. This method accomplishes considerable savings in memory—from as much as around 50× to 90×—and offers more than 180× computation speed-up and about 10× energy savings. All of these are gains that make Rosler extremely appropriate for learning on-device under resource-scarce situations, and available for a broad variety of scientific and industrial applications.

METHODS

A. *Standard Convolution*

A standard convolutional layer takes in a DF X DF X M feature map F and outputs a DG X DG X N feature map G where DF is the height and width of a square input feature map, M is the number of input channels, DG is the height and width of a square output feature map and N is the number of outputs.

Convolution of kernel K of size DK X DK X M on input feature map F gave output of size DG X DG X 1. When N such kernels are convolved on input, it gives an output volume G of size DG X DG X N where DG is width and height of a presumed square output.

The standard convolutional layer is parameterized by convolution kernel K of size DK X DK X M X N where DK is kernel dimension assumed to be square. The output feature map for standard convolution assuming stride one and padding is given by: Standard convolutions computational complexity:

1. single convolution: D_K X D_K X M
2. convolution of 1 kernel over input feature map F: D_G X D_G X D_K X D_K X M
3. convolution of N kernels over input feature map F: = $N\ X$ DG2 X DK2 $X\ M$

where the computational cost depends multiplicatively on the number of input channels M, the number of output channels N, the kernel size DK X DK, the output feature map G size DG X DG.

B. *Depth-wise Separable Convolution*

Depth-wise Separable Convolution address each standard convolution heavy computational terms and their interactions. First it uses depth-wise convolutions to break the interaction between the number of output channels and the size of the kernel and then it uses pointwise convolutions to combine the broken interactions in additive manner resulting in reduced multiplicative cost of standard convolution.

Depthwise separable convolution steps:

1. Depthwise convolution:

It takes as input a D_F X D_F X M feature map F where D_F is the width and height of assumed square input feature map. Convolution on kernel K of shape D_K X D_K X 1 over input feature map F of shape D_F X D_F X M results in output feature map G of size D_G X D_G X M where D_F is the width and height of input image, D_G is the width and height of output image and M is the number of input channels.

2. Pointwise Convolution

It takes input from depthwise convolution's output feature map as input feature map G of shape D_G X D_G X M.

The filter used in pointwise convolution is 1 X 1 X M which is basically 1X1 convolution operation over all M layers. If N such filters are applied on input feature map F, it results output feature map G of shape D_G X D_G X N.

Depthwise convolution computational complexity:

a. Depthwise operation (Filtering)

1. single convolution: DK X DK
2. convolution of 1 kernel over input feature map F for 1 channel: DG X DG X DK X DK
3. convolution of 1 kernel over input feature map

for M channels: DG X DG X DK X DK X M

b. Pointwise Operation (Combining)

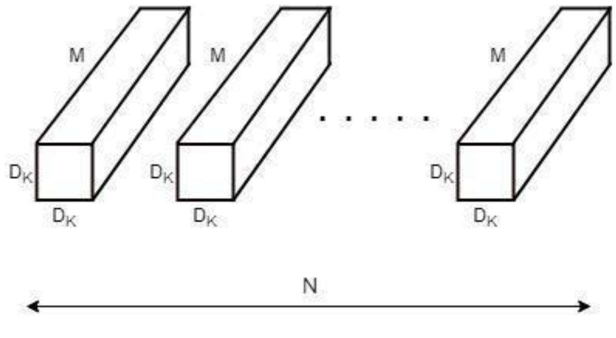
1. single convolution: M
2. convolution of 1 kernel over input feature map for 1 channel: DG X DG X M
3. convolution of N kernels over input feature

map for M channels: N X DG X DG X M

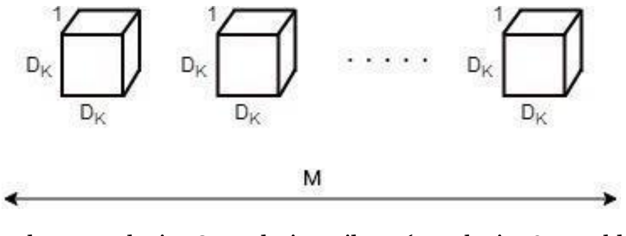
Total Computational Complexity:

$M\ X\ DGX\ DG\ X\ DK\ X\ DK + N\ X\ DG\ X\ DG\ X\ M$

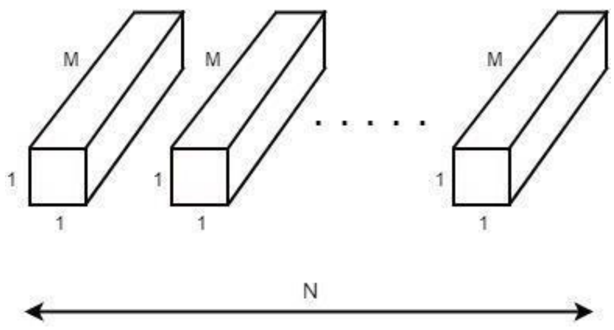
where the computational cost depends multiplicatively on the number of input channels M, the number of output channels N, the kernel size DK X DK, the output feature map G size DG X DG.



a. Standard Convolution Filters



b. Depthwise Convolution Filters (Depthwise Separable)



c. Pointwise Convolution Filters (Depthwise Separable)

C. Comparison of Standard Convolution and Depthwise Separable Convolution

By expressing convolution as a two-step process of filtering and combining we get a reduction in computation of:

=No. of mulit in depthwise separable convolutionNo. of mult in standard convolution

$$= \frac{MXD_KXD_KXD_GXD_G + D_GXD_GXMXN}{NXD_GXD_GXD_KXD_KXM}$$

= 1N+1DK2

MobileNet uses 3 X 3 depthwise separable convolutions which uses between 8 to 9 times less computation than standard convolutions at only a small reduction in accuracy as seen in Section 4

For e.g. Consider an output feature volume N = 1024 and DK of size = 3, plugging in the values in convolution expression, we can see

= 1N+1DK2= 11024+ 132

= 10339216=0.112

This indicates that depth separable convolution 8 – 9 times faster with respect to standard convolution in terms of computational complexity.

D. MobileNet Model (Depthwise separable CNN)

- Network Architecture and Training Process

The MobileNet architecture uses depthwise separable convolutions for all layers except the first layer which uses a standard convolutional operation. The total number of layers in MobileNet amounts to 28 when depthwise and pointwise convolutions operate as separate entities. Standard convolutional layers and MobileNet specialized convolutional layers show their structural differences in Figure 1. Measuring computational cost through Multiply-Add operations fails to provide a complete picture because practical implementation and efficient operation of these calculations must also be evaluated.

The training of MobileNet architectures used TensorFlow together with asynchronous gradient descent optimization in the same way as Inception V3 models. The training of MobileNet models did not include label smoothing or auxiliary classifier heads as implementation elements. The training process for larger models like Inception received limited image distortion from restricted aggressive cropping methods.

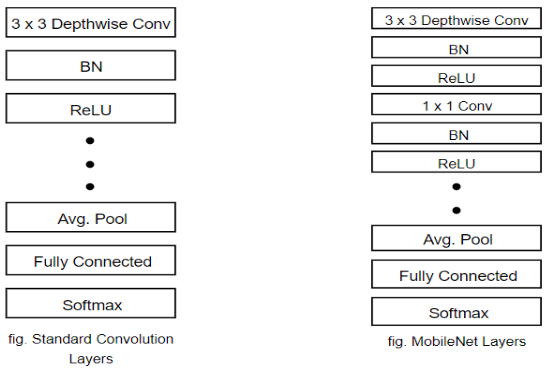


Figure 1. Standard Convolution and MobileNet Layers

- Width Multipler in MobileNet Model

Although the base MobileNet architecture is already optimized for low latency and compactness, certain applications may still require even smaller and faster models. To cater to such needs, MobileNet introduces a straightforward parameter known as the width multiplier (α\alphaα). This hyperparameter effectively reduces the width of the network at each layer in a uniform manner. The role of the width multiplier is to decrease the number of input and output channels across all layers, thereby reducing both memory and computation. The formula for computing the cost of a depthwise separable convolution with the width multiplier α\alphaα is given by:

$DK\ X\ DK\ X\ \alpha M\ X\ DF\ X\ DF + \alpha M\ X\ \alpha M\ X\ \alpha N\ X\ DF\ X\ DF$

Here, α\alphaα takes values in the range (0,1), commonly set as 1, 0.75, 0.5, or 0.25. By applying this multiplier, the overall number of operations and parameters is reduced approximately by a factor of α^2, significantly optimizing model performance for limited-resource environments.

- Resolution Multipler in MobileNet Model

In addition to controlling the width of the network, MobileNet employs a second hyperparameter known as the resolution multiplier (ρ). This parameter is used to reduce the input image resolution, thereby decreasing the spatial dimensions throughout the network and further lowering the computational load.

The computational cost when both the width multiplier α\alphaα and the resolution multiplier ρ\rhoρ are applied is expressed as:

$DK\ X\ DK\ X\ \alpha M\ X\ \rho DF\ X\ \rho DF + \alpha M\ X\ \alpha M\ X\ \alpha N\ X\ \rho DF\ X\ \rho DF$

Typical values for ρ are chosen so that the input image resolution becomes 224, 192, 160, or 128. This results in a quadratic reduction in spatial operations, approximately by a factor of ρ^2.

The following section will explore how adjustments in α\alphaα and ρ\rhoρ impact model accuracy versus resource efficiency, allowing MobileNet to be tailored for diverse deployment scenarios.

II. PROPOSED SYSTEM (SKETCH BASED RECOGNITION GAME)

A. INTRODUCTION

Sketches provide an instinctive method of communication which people have used since ancient times to effectively convey their ideas. Sketches present abstract representations which gather multiple concepts and thoughts into one representation.

The system needs to identify what the user intends to draw through their freehand sketch. The main objective involves teaching machines to recognize abstract human ideas expressed through sketches just as humans do when interpreting such visuals.

The system utilizes this understanding to develop interactive applications such as sketch-based games which require real-time interpretation of sketches to support gameplay that depends on abstract representation recognition.

The game basically is a set of this sketch tasks that a user needs to be completed in time and the neural network tries to guess the sketch. If it finds it to be true, user is rewarded with points based on what & how the sketch is drawn.

RESULTS

B. DATASET AND NEURAL NETWORK DETAILS

- DATASET:

We are using QuickDraw dataset open-sourced by google which is a huge collection of free hand human drawn sketches of real world objects. It contains about 5 million sketch images totalling its size to 3 TB of data but we are using on part of data about 70000 images for projects because of size constraints and low computational resources on Mobile Devices.

The data is categorized in raw-data and preprocessed data. The data is captured in time stamped vector format drawings and is available in formats - NDJSON, NPY, BIN. The dataset is freely available to be downloaded and used for development purposes.

- PRETRAINED MODEL

We are using MobileNet convolution model to train the images and generate pretrained model. TensorFlow comes packaged with great tools that you can use to retrain MobileNets without much code to write. The pretrained model uses parametric arguments to start training the dataset.

a. Model Training Parameters & Time Requirements:

- training steps = 8000
- architecture = MobileNet_1.0_224
- learning rate = 0.01
- train batch size = 512
- output graph = retrained_graph.pb

Table 1. Time Required for Training

Configuration	Time Required
Dual Core i3 CPU, 8GB RAM	28 Min 45 Sec
Quad Core i5 CPU, 8 GB RAM, 2GB 940MX GPU	20 Min 9 Sec
Octa Core Xeon CPU, 10 GB RAM	22 Min 17 Sec

b. Model Accuracy & Size, Comparison with state of art results:

The final training accuracy using MobileNet with 1.0 width multiplier with 224 resolution, learning rate 0.01 was 98 % and validation accuracy was 84.7 % which is more than The trained frozen graph model size was only 16.5 MB which is very small in size compared to standard CNN's frozen graph which was 98.4 MB. The accuracy observed in [17s] is 93.5%.

The trained model is used the application to recognise hand drawn sketches.

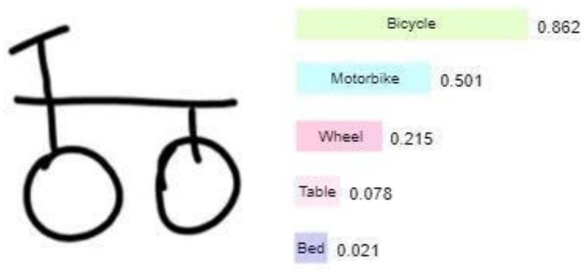


Figure 2. Classification of Sketch

C. GAMEPLAY

A task i.e. a sketch to be drawn is provided to the user that needs to be completed or drawn in the time provided for points.

Based on the what and how image is drawn for a given task, the app calculates a score for the image drawn and based on it and the level at which user is playing, an overall score is calculated for the task.

The task becomes harder and harder as you proceed through the levels, the harder task you solve, the more points you get.

The recognition threshold increases gradually and looks for more correct depiction of the given task gradually.

The Game provides two modes for playing:

- Normal Mode
- Arcade Mode
- Intime Mode

Normal Mode

In Normal Mode, a set of tasks ranging from easy to hard is provided and based on image drawn points are given as mentioned below. Once every task has been completed, an overall score is calculated which is additive of set of tasks. As the game progresses, task becomes harder.

Arcade Mode

In Arcade Mode, unlike Normal Mode, in this a set of tasks is given and based on drawn a score is given which is same across all tasks. If a task is not completed, the gameplay stops and the additive score is calculated as total score. Max 30 Points for each task based on how you draw the image.

In-Time Mode

In In-Time Mode, you have to complete as much tasks as possible in the given time which is 3 minutes.

DISCUSSION

Here we initially explore the impact of depthwise convolutions along with the option of shrinking by decreasing the width of the network instead of the number of layers. Also comparing Convolution Model Accuracy. Then we illustrate the trade-offs of shrinking the network according to the two hyper-parameters: width multiplier and resolution multiplier and compare to several well-known models. Also testing out learning rates for MobileNet and Inception. Android is being used as a testbed for trying out the experiments by creating pretrained models based on the tuned parameters in the sketch data from quickdraw by Google.

A. *MODEL OPTIONS*

Firstly, we show results in Figure 3 for MobileNet with depthwise separable convolutions compared to a model built with full standard convolutions and inception v3 model trained on 32000 28 x 28 images on Quad Core Processor with 8 GB RAM and 2 GB dedicated GPU.

We then provide results comparing with by adjusting width multiplier to less shallow models using fewer layers. The size, computation, and accuracy tradeoffs of MobileNet architecture shrinkage with respect to the width multiplier α are indicated in figure 4. The accuracy decays smoothly down until the architecture becomes too tiny at α = 0.25

Figure 5 illustrates the accuracy, computation and size for various resolution multipliers by training MobileNets with lower input resolutions. Accuracy declines smoothly with resolution.

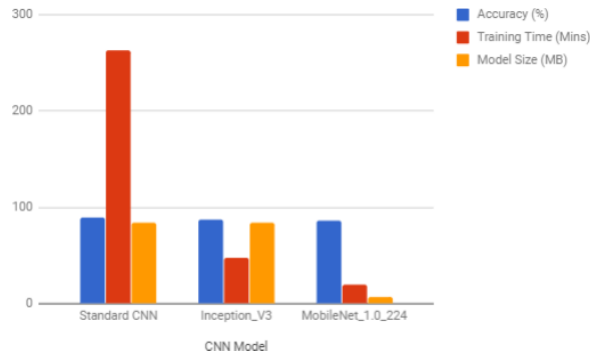


Figure 3. Comparison of CNN Models

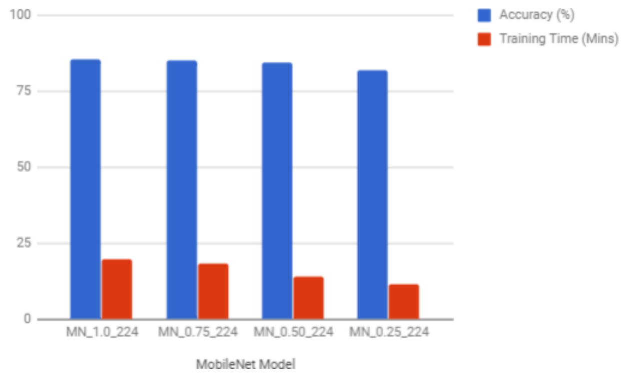


Figure 4. Comparison of Width Muliplier in MobileNet

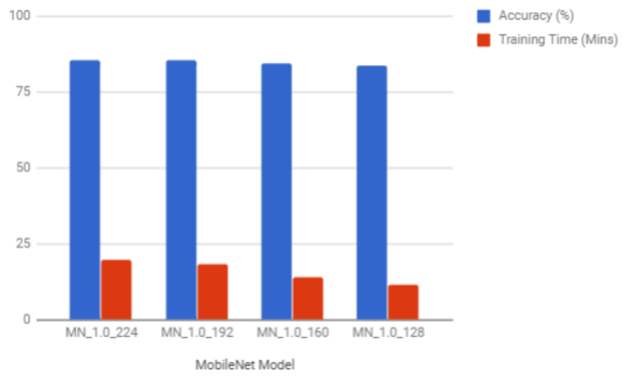


Figure 5. Comparison of Resolution Multipler in MobileNet

B. *LEARNING RATE:*

Next, we tried fiddling with learning rates and found a pretty big difference by adjusting the learning rates in terms of accuracy. We show the results obtained by adjusting the learning rates of MobileNet and Inception model below in Figure 6.

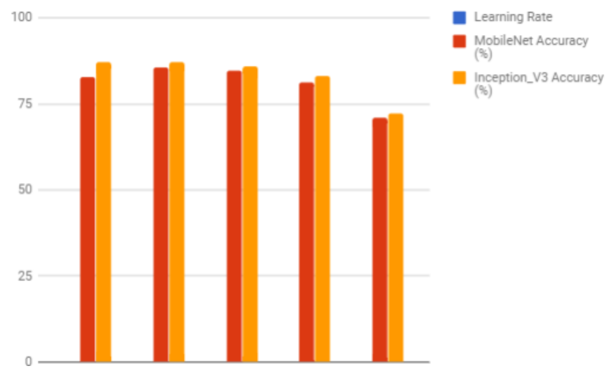


Figure 6. Comparison of Learning Rates on CNN Models

C. TRAINED MODEL

The pretrained model obtained by freezing the neural network generally in a protobuf or pb file format is used in mobile devices to classify images, text etc.

In figure 4, we can see how different convolution models produces frozen graphs (pretrained model) with different sizes. MobileNet model is the best when it comes to producing pretrained model which are very portable built for mobile devices.

Inception and Standard CNN produces pretrained model which has about 14 times larger than MobileNets pretrained model.

D. DATASET

We also trained Convolution Models on other datasets to analyze the performance effect on different datasets. We used data from ImageNet Dataset and Face Dataset. ImageNet containing of about 35,500 images of different real-life objects and Face Dataset containing about 10,000 images. The results indicated that MobileNet was much faster in training with using very less computational resources and time with minimum effect on accuracy. We got an accuracy of 78.4 % on ImageNet and 75.3 % on Face Dataset using MobileNet with trained model size of 16.5MB and 82.6 % on ImageNet and 82.3 % on Face Dataset using Inception V3 Model with 95.3 MB of trained model size.

CONCLUSION

In summary, we have performed both an experimental and theoretical study of the CNN models. Several changes have been identified and observed when adjusting the parameters of different convolution models. It was observed that there was a significant change in execution time when transitioning from standard convolution model to MobileNet model which uses depthwise separable convolution. The experiments suggest that MobileNet model for its portability of trained model, the time of execution and minimal effect on accuracy is suitable for the low computational resources available in mobile devices. We then demonstrated the application of MobileNet Model in a sketch recognition game using pretrained model for classifying sketches on Android and the results we obtained. The experimental results have been successfully interpreted.

REFERENCES

[1] Abdelhamied, Mahmoud M., Yasser M. Abd El-Latif, Fayed M. Ghaleb, and Ahmed MH Abdelfattah. "A Proposed Method to Acquire More Geometric Features from Hand-Drawn Sketches." *International Journal of Intelligent Engineering & Systems* 15, no. 5 (2022).

[2] Casper, Stephen, Carson Ezell, Charlotte Siegmann, Noam Kolt, Taylor Lynn Curtis, Benjamin Bucknall, Andreas Haupt et al. "Black-box access is insufficient for rigorous ai audits." In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pp. 2254-2272. 2024.

[3] Hussain, Muhammad, and Rahima Khanam. "In-depth review of yolov1 to yolov10 variants for enhanced photovoltaic defect detection." In *Solar*, vol. 4, no. 3, pp. 351-386. MDPI, 2024.

[4] Mihai, Stefan, Mahnoor Yaqoob, Dang V. Hung, William Davis, Praveer Towakel, Mohsin Raza, Mehmet Karamanoglu et al. "Digital twins: A survey on enabling technologies, challenges, trends and future prospects." *IEEE Communications Surveys & Tutorials* 24, no. 4 (2022): 2255-2291.

[5] Walia, Guneet Kaur, Mohit Kumar, and Sukhpal Singh Gill. "AI-empowered fog/edge resource management for IoT applications: A comprehensive review, research challenges, and future perspectives." *IEEE Communications Surveys & Tutorials* 26, no. 1 (2023): 619-669.

[6] Song, Binyang, Scarlett Miller, and Faez Ahmed. "Attention-enhanced multimodal learning for conceptual design evaluations." *Journal of Mechanical Design* 145, no. 4 (2023): 041410.

[7] Terven, Juan, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas." *Machine learning and knowledge extraction* 5, no. 4 (2023): 1680-1716.

[8] Fan, Judith E., Wilma A. Bainbridge, Rebecca Chamberlain, and Jeffrey D. Wammes. "Drawing as a versatile cognitive tool." *Nature Reviews Psychology* 2, no. 9 (2023): 556-568.

[9] Moser, Brian B., Arundhati S. Shanbhag, Federico Raue, Stanislav Frolov, Sebastian Palacio, and Andreas Dengel. "Diffusion models, image super-resolution, and everything: A survey." *IEEE Transactions on Neural Networks and Learning Systems* (2024).

[10] Sain, Aneeshan, Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Subhadeep Koley, Tao Xiang, and Yi-Zhe Song. "Clip for all things zero-shot sketch-based image retrieval, fine-grained or not." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2765-2775. 2023.

[11] Guo, Yuhao, Yicheng Li, Shaohua Wang, Kecheng Sun, Mingchun Liu, and Zihan Wang. "Pedestrian multi-object tracking combining appearance and spatial characteristics." *Expert Systems with Applications* (2025): 126772.

[12] P. Xu, T. M. Hospedales, Q. Yin, Y. -Z. Song, T. Xiang and L. Wang, "Deep Learning for Free-Hand Sketch: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 1, pp. 285-312, 1 Jan. 2023, doi: 10.1109/TPAMI.2022.3148853..

[13] L. Xuan, K. -F. Un, C. -S. Lam and R. P. Martins, "An FPGA-Based Energy-Efficient Reconfigurable Depthwise Separable Convolution Accelerator for Image Recognition," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 10, pp. 4003-4007, Oct. 2022, doi: 10.1109/TCSII.2022.3180553.

[14] P. Xu, C. K. Joshi and X. Bresson, "Multigraph Transformer for Free-Hand Sketch Recognition," in IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 10, pp. 5150-5161, Oct. 2022, doi: 10.1109/TNNLS.2021.3069230.

[15] Moutsis, Stavros N., Konstantinos A. Tsintotas, Ioannis Kansizoglou, and Antonios Gasteratos. 2023. "Evaluating the Performance of Mobile-Convolutional Neural Networks for Spatial and Temporal Human Action Recognition Analysis" Robotics 12, no. 6: 167. https://doi.org/10.3390/robotics12060167.

[16] Shaukat Hayat, Kun She, Muhammad Mateen and Yao Yu, "Deep CNN-based Features for Hand-Drawn Sketch Recognition via Transfer Learning Approach" International Journal of Advanced Computer Science and Applications(IJACSA), 10(9), 2019. http://dx.doi.org/10.14569/IJACSA.2019.0100958

[17] Huynh, Viet-Tham, Trong-Thuan Nguyen, Tam V. Nguyen, and Minh-Triet Tran. "MobileNet-SA: Lightweight CNN with Self Attention for Sketch Classification." In *Pacific-Rim Symposium on Image and Video Technology*, pp. 110-123. Singapore: Springer Nature Singapore, 2023.

[18] J. Ahmad, K. Muhammad, S. I. A. Shah, A. K. Sangaiah, and S. W. Baik, "Partially shaded sketch-based image search in real mobile device environments via sketch-oriented compact neural codes," Journal of Real-time Image Processing, vol. 16, no. 1, pp. 227–240, Feb. 2019, doi: 10.1007/S11554-018-0784-X. Available: https://dl.acm.org/doi/abs/10.1007/s11554-018-0784-x

[19] S. Elliott, "Light-SRNet: a lightweight dual-attention feature fusion network for hand-drawn sketch recognition," Journal of Electronic Imaging, vol. 32, no. 01, Jan. 2023, doi: 10.1117/1.jei.32.1.013005

[20] B. Li, B. Li, P. Chen, L. Hongfu, W. Guo, W. Guo, X. Cao, J. Du, C. Zhao, J. Zhang, and J. Zhang, "Random sketch learning for deep neural networks in edge computing," Nat Comput Sci 1, 221–228 (2021). https://doi.org/10.1038/s43588-021-00039-6