Introduction to Machine Learning
Mini Project 3

Gian Favero[1] **Hieu Thien Hoang**[2] **Maxime Favreau-Vachon**[1]
[1] McGill University [2]École de Technologie Supérieure
gian.favero@mila.quebec
thien.hoang@mail.mcgill.ca
maxime.favreau-vachon@mail.mcgill.ca

Abstract

This paper presents a comprehensive exploration of image analysis techniques for classification of the EMNIST dataset, designed for a 10-class classification task. We introduce a novel ensemble model of existing state-of-the-art CNN architectures to amplify feature extraction capabilities and achieve well-generalizable and robust performance on the unseen Kaggle test set. The ensemble model, comprising VGG-5 and SpinalNet, demonstrates superior performance compared to each individual model in accuracy. Further, remarkable alignment with the paper's reported results on the EMNIST dataset confirms the fidelity of our model reproduction, underlining the effectiveness of our tailored approach for the specified classification task.

1 Introduction

In the evolving landscape of machine learning and neural network architectures, this project undertakes a comprehensive exploration and replication of the VGG-5 and SpinalNet models introduced in a seminal paper [1]. Our investigation is centered around a dataset comprising qualitative images, each showcasing a Japanese numeral alongside characters sourced from the EMNIST dataset, tailored for a classification task with 10 distinct classes. The dataset features a batch of 60,000 grayscale images, each sized 28x28 pixels. Notable preprocessing techniques include pixel value normalization to a standardized range, promoting fairness and stability in training. Additionally, random translations and rotations diversify the training dataset, mitigating overfitting and enhancing the model's adaptability to diverse inputs.

In this project, we aim to faithfully reproduce the model proposed in the cited paper where they draw inspiration from the well-established VGG16 [2] architecture and integrate the pioneering SpinalNet layer, thereby adapting the more concise VGG5 model [1]. Our distinctive contribution introduces an additional layer of sophistication by presenting an ensemble model that merges VGG-5 and VGG-5 with SpinalNet. This innovative approach underscores superior performance when compared to the individual models, demonstrating the efficacy of our refined ensemble strategy.

2 Datasets

In this project, our dataset consists of qualitative images, with each image featuring a Japanese numeral alongside additional characters sourced from the EMNIST dataset. The dataset is specifically curated for a classification task encompassing 10 distinct classes. The training image data is represented as torch.Size([60000, 1, 28, 28]), indicating a batch of 60,000 grayscale images, each with dimensions 28x28 pixels. This tensor structure is designed for efficient processing in tasks such as neural network-based classification.

2.1 Class distribution

Figure 1 illustrates the class distribution in each dataset, showcasing a balanced distribution where 10% of the images are allocated to each class.

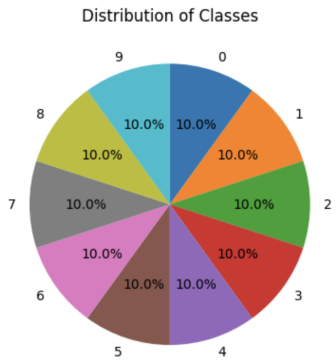

Figure 1: Distribution of the ten classes

2.2 Preprocessing

For preprocessing the dataset, we implement several essential techniques to enhance the robustness and effectiveness of our model. First, we made sure that the images' pixel values are normalized, ensuring their scaling to a standardized range, typically [0,1]. This normalization fosters fairness across all images, preventing disparities where images with varying pixel ranges contribute unevenly to the total loss during training and creating an equitable learning environment. This strategic normalization not only contributes to stability during training but also facilitates a more consistent and effective learning process, accommodating variations in pixel intensity and promoting balanced convergence across diverse image characteristics [3].

Additionally, we introduce random translations to training images, which involve shifting the image in horizontal and vertical directions. We incorporate random rotations of training images, introducing slight rotations to diversify the dataset. Performing data augmentation plays a pivotal role in preventing a neural network from learning irrelevant features. This strategic approach ensures that the model is exposed to a variety of transformed inputs, preventing it from fixating on specific details by exposing it to variations in object placement and enhancing its ability to discern relevant patterns. Consequently, this contributes to better model performance on unseen data by promoting robust learning and reducing the risk of overfitting [4]. The data was partitioned into an 80-20 training and validation split, for training and hyperparameter tuning, with a held-out test set comprising of 10,000 images to be submitted to Kaggle.

3 Proposed Approach

Our journey into the realm of convolutional neural networks initially led us to the popular VGG16 architecture, known for its exceptional performance in computer vision tasks. As we delved deeper into our exploration, we aimed to simplify the intricacies of VGG16 while preserving its effectiveness. Inspired by a very promising paper in the field [5], we embarked on the task of adapting VGG-5, a more concise variant, with the innovative SpinalNet layer. The objective was to enhance feature extraction capabilities and model performance. Taking a step further, our approach involved forming an ensemble of both VGG-5 and SpinalNet, dynamically selecting the model with the highest confidence based on log-probability. This strategy aimed to harness the strengths of both architectures, achieving a robust solution for our specific objectives.

3.1 Classification Models

3.1.1 VGG16

VGG16, short for Visual Geometry Group 16, is a prominent convolutional neural network renowned for its excellence in computer vision tasks. Developed as an object detection and

classification algorithm, VGG16 stands out as one of the most effective models in image classification, boasting an impressive 92.7% accuracy on ImageNet [2]. What sets VGG16 apart is its emphasis on depth, featuring 16 weight layers despite having 21 layers in total. Notably, the architecture incorporates thirteen convolutional layers, five Max Pooling layers, and three Dense layers.

The uniqueness of VGG16 lies in its consistent use of 3x3 convolution filters with a stride of 1, always employing the same padding and utilizing 2x2 max-pooling layers with a stride of 2. The convolution and max-pooling layers are meticulously organized throughout the architecture, with each convolutional layer having a distinct number of filters, ranging from 64 to 512. With a focus on simplicity and effective feature extraction, VGG16 has become a widely used model, especially in applications leveraging transfer learning [2].

3.1.2 VGG-5

The paper, [5], introduces a reduced model of VGG-16 referred to as VGG-5. While VGG-16 is a deeper and more complex model with four blocks of convolutional layers ranging from 64 to 512 filters, making it suitable for tasks with higher complexity, VGG-5 consists of four blocks, each containing convolutional layers, batch normalization, ReLU activation, and max-pooling layers. The convolutional layers in these blocks have 32, 64, 128, and 256 filters, respectively. This complete architecture can be seen in Figure 2. The reduced complexity of VGG-5 offers advantages in scenarios with less demanding tasks and strikes a balance between complexity and computational efficiency.

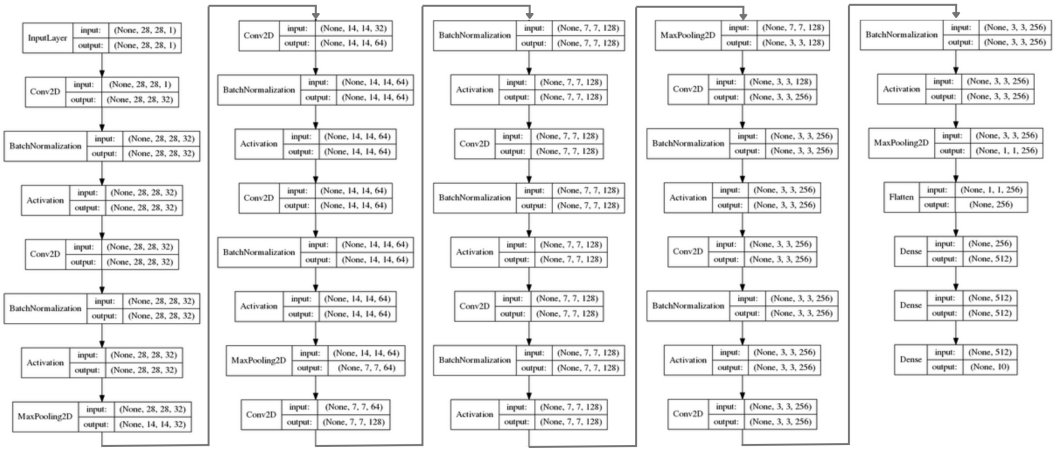

Figure 2: VGG5 Architecture

3.1.3 VGG-5 with SpinalNet

The paper [5] also introduces SpinalNet, a novel neural network architecture, with the primary aim of enhancing performance while minimizing computational overhead. The proposed structure, characterized by gradual and repetitive input capabilities, enable neural networks to achieve promising results using a reduced number of parameters. The network architecture comprises input sub-layers, intermediate sub-layers, and an output layer, where input data is distributed across multiple hidden layers.

Illustrated in Figure 3, each intermediate sub-layer accommodates two neurons per hidden layer, with the flexibility for users to adjust the number of intermediate neurons. Both the number of intermediate neurons and inputs per layer are intentionally kept small to minimize computational complexity. This design, although potentially leading to an under-fit shape, fosters interconnectivity between layers, allowing crucial features to influence the output across various hidden layers. The intermediate sub-layers integrate a nonlinear activation function, while the output layer employs a linear activation function. Input values are divided into three rows, cyclically assigned to different hidden layers, enhancing the network's adaptability and capturing diverse features throughout the hierarchical structure. The study delves into the integration of SpinalNet as the fully connected layer within the VGG-5 network, showcasing state-of-the-art performance across four MNIST datasets.

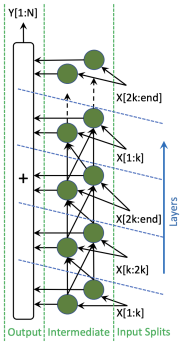

Figure 3: SpinalNet Architecture

3.1.4 Ensemble Classifier

In our project, our primary objective is to implement the VGG-5 and SpinalNet models and reproduce the success reported in previous works [5]. To further explore the predictive capacity of models, we have introduced a novel ensemble approach. This ensemble consists of both the standalone VGG-5 model and the adapted VGG-5 with SpinalNet as the fully connected layer. Predictions are made via a "most-confident" delegation taken from the output layer of each model. By harnessing the complementary strengths and diverse features learned by each model, our ensemble strategy aims to improve overall performance, mitigate overfitting, and enhance the model's versatility across various scenarios.

4 Results

We run all experiments on a single 16 GB NVIDIA V100 GPU for 150 epochs. A randomized 80-20 partition is used to separate the data into training and validation splits for each experiment. Cross-entropy loss is used as as the training objective, and validation accuracy is used to evaluate the performance of each model.

4.1 Hyperparameter Tuning for VGG-5

We begin with an implementation of the base VGG-5 model and assess performance with various batch size and activation functions (Table 1). The results show that a VGG-5 model trained with a batch size of 64 and ReLU activation functions in each layer provide the best performance on the validation set.

	Batch size			Activation function		
	16	32	64	ReLU	LeakyReLU	tanh
Training Loss	0.0355	0.0227	0.0650	0.0650	0.0854	0.2284
Validation Loss	0.0683	0.0784	0.0706	0.0706	0.0799	0.1113
Validation Accuracy	98.18%	98.00%	**98.20%**	**98.20%**	97.92%	96.58%

Table 1: Batch size and activation function tunings

We extend our analysis to the optimizer selected for training. A VGG-5 model is trained with various optimizers and its validation performance is summarized in Table 2. We find that the best performance is achieved with an Adam optimizer.

	Optimizer			
	Adam	AdamW	NAdam	SGD
Training Loss	0.0650	0.1424	0.0379	0.8203
Validation Loss	0.0706	0.0932	0.0841	0.2372
Validation Accuracy	**98.20%**	97.61%	97.87%	92.93%

Table 2: Optimizer tuning

4.2 VGG-5, VGG5-SpinalNet, and Ensemble Models

We progress to training our implementations of VGG-5, VGG5-SpinalNet, and ensemble models. A learning rate of 0.001 is used initially, which is decreased exponentially after a warm-up period of 20 epochs. Performance of each model is reported in Table 3.

	VGG-5	**VGG5-SpinalNet**	**Ensemble**
Training Loss	0.0650	0.0712	-
Validation Loss	0.0706	0.0895	**0.0612**
Validation Accuracy	98.20%	97.64%	**99.71%**
Kaggle	94.73%	95.00%	**95.47%**

Table 3: Model Accuracies

5 Discussion and Conclusion

The observed performance differences between VGG-5 and VGG-5 with SpinalNet, as well as the superiority of the ensemble, can be attributed to several factors. While the simplicity of VGG-5 could lead to faster convergence and exhibits lower training and validation losses, suggesting a better fit to the training data, it might lack the capacity to capture more intricate patterns that are present in the Kaggle test dataset. On the other hand, VGG-5 with SpinalNet, despite having slightly higher losses, showcases enhanced generalization capabilities, as evidenced by a better Kaggle score on new data. The SpinalNet modification likely introduces features that contribute to improved performance on previously unseen examples. The ensemble model take advantage of the strengths of both architectures, leveraging their complementary features. The ensemble's exceptional validation accuracy and Kaggle score highlight the effectiveness of combining diverse models, demonstrating the power of ensemble learning in achieving superior performance across various datasets and tasks. The ensemble's success reinforces the idea that, by aggregating predictions from multiple models, it becomes possible to mitigate weaknesses and enhance overall predictive accuracy.

Additionally, it is noteworthy that our model's performance aligns closely with the results reported in the paper. In the original study, VGG-5 achieved an average accuracy of 95.71%, while VGG-5 with SpinalNet exhibited a slightly higher accuracy of 95.79% on the EMNIST dataset of letters. Our replicated models' comparable performance on this dataset suggests that the adaptations and modifications made during the reproduction process have successfully captured the essence of the proposed approach. This alignment with the paper's reported results reinforces the credibility and effectiveness of our model in tackling similar tasks and datasets.

To improve our model, we might explore strategies like adjusting hyperparameters and experimenting with different architectures. Given more time, refining image quality, specifically addressing observed blurriness, could be considered. Investigating valuable sharpening techniques, with some studies favoring the Median Filter for preserving symbol edges and reducing noise, is noteworthy. Various sharpening methods involve creating a mask from an "unsharp" negative image, then combining it with the original image for a sharper version. Additionally, exploring attribute reduction techniques for efficient data representation which consists of dividing the image into blocks and deriving attributes, like the mean of each block. The effectiveness of these sharpening techniques, demonstrated in [6], is proven to enhance model performance.

In summary, our findings underscore the nuanced trade-offs in performance between VGG-5, VGG-5 with SpinalNet, and the ensemble, showcasing the latter's capacity to harness their complementary strengths. The close alignment with the paper's reported results on the EMNIST dataset reaffirms the fidelity of our model reproduction. Looking ahead, potential improvements involve exploring image quality refinement and attribute reduction techniques.

6 Statement of contributions

Every team member contributed equally to the construction of the experiments run and to this report.

References

[1] M. Kweon, "mnist-competition," 04 2017. [Online]. Available: https://github.com/kkweon/mnist-competition

[2] G Rohini, "Everything you need to know about vgg16," https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918, 2021.

[3] Fredrick Adhinga, "Getting started with image preprocessing in python," https://www.section.io/engineering-education/image-preprocessing-in-python/, 2021.

[4] Yash Chauhan, "Data augmentation in image classification models," https://medium.com/international-school-of-ai-data-science/increase-the-performance-of-image-classification-models-b466e1ae3101, 2023.

[5] H. M. D. Kabir, M. Abdar, S. M. J. Jalali, A. Khosravi, A. F. Atiya, S. Nahavandi, and D. Srinivasan, "Spinalnet: Deep neural network with gradual input," 2022.

[6] S. Chen, R. Almamlook, Y. Gu, and L. Wells, "Offline handwritten digits recognition using machine learning," in *Proceedings of the international conference on industrial engineering and operations management*, 2018, pp. 274–286.