Implementation of VGG architecture

1 Abstract

In our mini-project, we replicated key findings from the "Very Deep Convolutional Networks for Large-Scale Image Recognition" [6] paper. Our objective was to examine the impact of network depth on image recognition [1] accuracy. Utilizing the Visual Geometry Group (VGG) [7] model family, we observed notable improvements in performance with network depths of 11-19 layers and compact 3x3 convolution kernels. This approach proved significantly more effective than previous models, emphasizing the benefits of deep networks in image recognition. Our project not only validates the original paper's insights but also demonstrates the practical applications and advantages of deep convolutional networks [5] in advancing the field of large-scale image recognition.

2 Introduction

The realm of large-scale image and video recognition has seen significant strides propelled by Convolutional Networks (ConvNets) [8], especially with the aid of resources like ImageNet and advancements in computational infrastructure such as GPUs and distributed clusters . ConvNets have become pivotal in computer vision, prompting efforts to refine their architectures for heightened accuracy. Notable refinements, including adjustments in receptive window sizes and strides within the initial convolutional layer, alongside advancements in dense network training and comprehensive image testing across multiple scales, marked significant progress in ILSVRC2013 submissions.This study focuses on a crucial aspect of ConvNet architecture—depth. Specifically, it investigates various iterations of VGG models, spanning 11 to 19 layers, shifting from larger kernels to smaller ones to explore the impact of deep networks using these smaller kernels. Model performance evaluation utilizes a subset of the ImageNet database, revealing that the deeper network, VGG-19, emerged as the most effective performer. Moreover, the study delves into the influence of normalization layers on performance.Seeking robustness, the algorithm's performance undergoes scrutiny across diverse combinations of train and test image sizes, incorporating scale jittering. Instead of training on a fixed image size, employing scale jittering notably enhances results. The findings underscore that deeper networks outperform those utilizing larger kernels, unveiling a pivotal revelation in ConvNet architecture exploration.

3 Datasets

The database used in this study is derived from the ImageNet dataset [2], comprising 9469 images in train set and 3925 images in test set, categorized into 10 distinct classes. The original ImageNet dataset, housing over 14 million images across 1,000 classes, was constrained due to hardware limitations. The selected classes in our database are as follows: "tench", "English springer", "cassette player", "chain saw", "church", "French horn", "garbage truck", "gas pump", "golf ball", and "parachute".All images within our database maintain a resolution of 320 pixels, meeting the base standard. However, for experimental purposes, certain images underwent rescaling in alignment with specific experimental requisites. The distribution of images per class is visually represented through a bar plot Figure 1 and Figure 2, illustrating the frequency distribution across the aforementioned 10 classes.

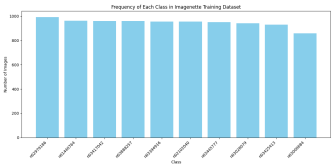

Figure 1: Train Data

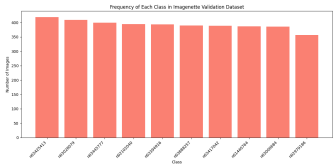

Figure 2: Validation Data

4 Architecture

Our ConvNet architecture, inspired by VGG [7], processes fixed-size 224 × 224 RGB images during training. We utilize 3 × 3 convolutional layers and occasionally integrate 1 × 1 convolution filters for linear transformations. The stride is fixed at 1 pixel, maintaining spatial resolution, and max-pooling with a 2 × 2 pixel window and a stride of 2 is applied strategically. Following the conv. layers, our design mirrors VGG with a stack of Fully-Connected (FC) layers [4]. The first two FC layers have 4096 channels each, and the third layer facilitates 1000-way ILSVRC classification. Like VGG, rectification non-linearity is applied to all hidden layers, while Local Response Normalisation (LRN) is omitted based on VGG's findings. Parameters for LRN, when applicable, align with Krizhevsky et al. (2012) [3].

- VGG A/VGG A-LRN: This particular variant of VGG features an 11-layer model, making it the lightest among all available variants. The A-LRN version includes a normalization layer inserted between the layers to evaluate its effectiveness
- VGG B: This variant is a 13-layer model with few extra convolution layers compared to the A model.
- VGG C: This VGG variant is a 16-layer model, renowned for its widespread usage. It is a deep model with strong generalization capabilities on test data. In this version, the convolution kernel size is 1x1.
- VGG D: This variant is a 16-layer model, akin to the C model, but it features a larger convolution kernel of size 3x3. This design enables it to more effectively capture local dependencies compared to the C model.
- VGG E: This variant, with its 19-layer structure, stands as the deepest among all VGG models. Its performance surpasses that of all other variants.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 3: Architecture

5 Experiments and Results

The dimensions of the training set (S) and test set (Q) were systematically adjusted within the range of 256px to 384px, and corresponding results were acquired. In accordance with the original paper, the variation in image size spanned from 256px to 512px. However, due to constraints related to memory capacity, the maximum permissible image size was constrained to 384px.
The models A, B, and C were trained by optimizing categorical cross-entropy loss through the Adam optimizer. Training involved regularization via a dropout (dropout ratio set at 0.5) applied to the initial two fully-connected layers. Learning was halted after 74 epochs with a learning rate of 1e-5. In contrast, models D and E utilized pre-trained Keras models for training.

5.1 Single Scale Evaluation

In the single-scale evaluation, the performance of the model was assessed using varying test image sizes. Initially, the test image size was set to be equal to the training image size (Q=S) for a fixed S, and Q was set to 0.5(Smin+Smax) for $S \in [256,512]$. Notably, the application of local response normalization did not yield significant improvements in accuracy, as evidenced by comparable results between models A and A-LRN. Upon increasing the model's depth from 11 to 19 layers, a discernible decrease in validation error was observed, with model D featuring 19 layers demonstrating superior performance. It is noteworthy that, despite both models C and D having an equal number of layers (16), model C, utilizing a 1x1 kernel, performed less favorably than model D, which employed a 3x3 kernel. This observation underscores the importance of capturing spatial context through convolutional filters with non-trivial receptive fields, as model D outperformed model C.

Furthermore, the incorporation of scale jittering, where $S \in$[256,512], proved to be beneficial. This approach outperformed training the model on a single scale, emphasizing the positive impact of scale diversity. Detailed results are presented in Table 1.

Model	Image size		Top 1% val accuracy	Top 5% val accuracy
	Train(S)	Test(Q)		
A	256	256	88.7%	98%
A-LRN	256	256	88.5%	98%
B	256	256	66.3%	90.2%
C	256	256	81.2%	92.2%
	384	384	82.3%	99%
	[256;512]	320	82.5%	99%
D	256	256	94.8%	99%
	384	384	96.4%	99.6%
	[256;512]	320	96.7%	99.7%
E	256	256	94%	99.1%
	384	384	96.8%	99.7%
	[256;512]	320	97.1%	99.6%

Table 1: Result of Single Scale evaluation

5.2 Multi Scale Evaluation

The experimental evaluation conducted a multi-scale assessment by testing the model across diverse rescaled versions of the test image sizes (Q). To accommodate varying train and test image sizes, a transformation was applied to the Fully Connected (FC) layers, converting them into convolutional layers. This conversion entailed utilizing filters of sizes 7x7, 1x1, and 1x1 for the three FC layers, respectively. Additionally, a global max pooling layer was introduced at the conclusion of the architecture. This strategic addition aimed to eliminate result size dependency on image size, establishing reliance instead on the number of channels and classes within the model.

Recognizing the detrimental impact of substantial discrepancies between training and testing scales on performance, models trained with a fixed training set size (S) were assessed over three test image sizes in proximity to the training scale: Q = {S - 32, S, S + 32}. Concurrently, employing scale jittering during training allowed the network to adapt to a broader spectrum of scales during testing. Consequently, the model trained with variable $S \in$[Smin; Smax] was evaluated across a wider range of sizes: Q = {Smin, 0.5(Smin + Smax), Smax}.In line with our findings from the Single-Scale Evaluation, models D and E exhibited superior performance, achieving 96% and 97% accuracy, respectively. Leveraging scale jittering, however, notably enhanced performance to 98%. It is noteworthy that our experimentation utilized a subset of the dataset comprising only 10 classes, compared to the original dataset containing 1000 classes. This refinement resulted in more finely tuned results, yielding improved accuracy compared to the baseline presented in the original paper. The detailed results are shown in Table 2.

Model	Image size		Top 1% val accuracy	Top 5% val accuracy
	Train(S)	Test(Q)		
B	256	224,256,288	65%	94.2%
C	256	224,256,288	78%	96%
	384	352,384	81.7%	96%
	[256;512]	256,320,384	82%	97%
D	256	224,256,288	94.8%	99%
	384	352,384	97%	99%
	[256;512]	256,320,384	98%	99%
E	256	224,256,288	93.6%	99%
	384	352,384	96%	99%
	[256;512]	256,320,384	98%	99%

Table 2: Result of Multi Scale evaluation

5.3 Multi Crop Evaluation

In this section, we evaluate the performance of the ConvNet [8], specifically employing a multicrop evaluation technique. We consider 4 variations of VGG [7] where the finetuning of the model is done using the training data of image size $Q = [256, 512]$.

In the test side, we evaluated the different test scales $S = 256, 384, 512$.. This approach allows us to understand how the model performs with varying image scales, which is crucial for its applicability in real-world scenarios where image sizes can differ significantly. We mentioned the results in the below table.

Model	Image size		Top 1% val accuracy
	Train(S)	Test(Q)	
E	256	256	64.6%
	256	384	43.2%
	256	512	40.1%
D	256	256	60.2%
	256	384	39.8%
	256	512	35.6%

Table 3: Result of Multi Crop evaluation

5.4 ConvNet Fusion

In this section, we assess the effectiveness of an ensemble approach where the outputs of multiple models are integrated by averaging their softmax class posteriors. This technique enhances the overall system's performance due to the combined strengths of individual models. The averaging of class posteriors from various models leads to a more robust and reliable prediction, as it leverages the unique capabilities and insights of each model. This method effectively mitigates individual model biases and errors, resulting in improved accuracy and reliability in the final predictions. We have taken the ensemble of VGG-16 and VGG-19 models. We got an Top 1% Accuracy: 0.95 and Top 5% Accuracy: 0.99.

6 Ablation Study

6.1 Experiments with different kernel size

In this study, we investigated the impact of replacing the fully connected layers [4]in the model with convolutional layers of varying filter sizes during the evaluation of rescaled versions of the test image(Q). The fully connected layers were converted to convolutional layers, with the initial layer utilizing a 7x7 filter and the last two layers employing 1x1 filters. Specifically, we conducted experiments to substitute the 7x7 filter layer with filters of sizes 5x5 and 3x3.The utilization of a 5x5 filter yielded promising results, demonstrating an accuracy of 94% for VGG-16 model and 92% for VGG-19 model, closely resembling the accuracy achieved by the model utilizing a 7x7 filter. This observation suggests that the 5x5 filter effectively captured the spatial context within the images. Additionally, employing smaller filter sizes proved beneficial in reducing the number of parameters without significant compromise in accuracy.However, upon replacing the 7x7 filter with a 3x3 filter, a drastic reduction in accuracy to 67% was observed. This notable decline in performance could be attributed to the model's struggle to capture larger and more intricate features within the images, potentially leading to underfitting.The experiments were rigorously conducted on both the VGG-16 and VGG-19 models, and a comprehensive breakdown of the results is provided in the Table 4.

Model	7 ×7		5 ×5		3 ×3	
	Train	Validation	Train	Validation	Train	Validation
VGG-16	94.8%	94%	88.8%	94.9%	68%	63%
VGG-19	96%	96%	90%	92%	66%	58%

Table 4: Accuracy with different kernel size in last layers

6.2 VGG model with 21 and 23 layers

In this section, we incorporated deeper models, specifically the VGG-21 and VGG-23 models, as illustrated in Figure 6. Consistent with the findings in the original paper, we observed that models deeper than VGG-19 tend to overfit and exhibit poor performance. Our results aligned with this observation, depicting overfitting of the model and notably low validation accuracy, as depicted in the Figure 4 and Figure 5.

7 Discussion

The findings from our mini-project, which replicated key aspects of the "Very Deep Convolutional Networks for Large-Scale Image Recognition" paper, illuminate several critical insights into ConvNet architecture and performance. Our study's

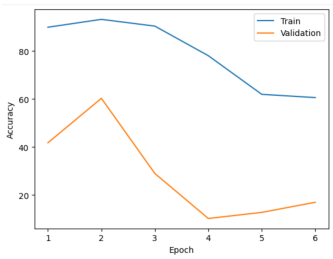

Figure 4: VGG-21 model

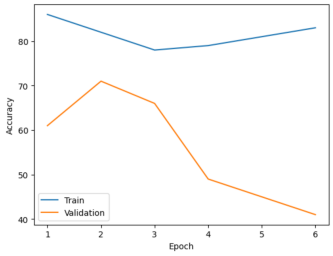

Figure 5: VGG-23 model

primary focus on the impact of network depth on image recognition accuracy has yielded significant results, particularly emphasizing the effectiveness of deep networks using compact 3x3 convolution kernels.

One of the most notable revelations from our experiments is the superior performance of deeper networks, specifically those ranging from 11 to 19 layers, in comparison to shallower architectures. This enhancement in performance with increased depth challenges previous norms and highlights the advantages of deep learning in image recognition tasks. Furthermore, our experiments underscored the effectiveness of scale jittering, which significantly improved model performance. This finding suggests that training models on varied image scales can contribute to a more robust and adaptable network, capable of handling diverse image resolutions effectively.

Interestingly, our experiments also revealed that the size of convolution kernels plays a crucial role in model performance. While smaller kernels (3x3) were generally more effective, especially in deeper networks like VGG D and VGG E, the drastic reduction in performance with even smaller kernels (1x1) in certain models suggests a complex interplay between kernel size, network depth, and overall architecture.

Moreover, the exploration of very deep models (VGG-21 and VGG-23) provided an important boundary condition to our findings: there is a threshold beyond which additional depth may lead to overfitting and reduced performance. This observation aligns with the principles of Occam's razor in model design, indicating that while complexity can enhance performance, excessive complexity without adequate regularization or data support can be detrimental.

8 Conclusion

Our project successfully replicated and extended key findings from a seminal paper in ConvNet architecture, contributing valuable insights into the field of large-scale image recognition. By exploring various VGG model iterations and manipulating factors such as network depth, kernel size, and training scale variability, we have deepened the understanding of factors influencing ConvNet performance.Our results not only validate the original paper's insights but also demonstrate the practical applications and advantages of deep convolutional networks in advancing large-scale image recognition. They highlight the importance of optimal network depth, effective kernel size, and the potential benefits of scale jittering in training. These findings contribute to the ongoing development of more efficient and accurate image recognition systems and pave the way for future research in the field of deep learning and computer vision.

9 Statement of Contribution

Shwetali Shimangaud's contributions to the research paper involved conducting experiments focused on both single and multi-scale evaluations. She also delved into the exploration of more intricate VGG-21 and VGG-23 models. Dheeraj Vattikonda contributed by implementing Multi-crop experiments and performing ConvNet fusion experiments. Chaitanya Tekane provided support in composing the report.

Due to limitations in computational resources, we were unable to fully train the models or conduct extensive fine-tuning. However, we endeavored to thoroughly explore every aspect of the paper and experimented with various elements that were feasible within our constraints

	ConvNet Configuration	
E	**F**	**G**
19 weight layers	**21 weight layers**	**23 weight layers**
Input image		
conv3-64	conv3-64	conv3-64
conv3-64	conv3-64	conv3-64
maxpool		
conv3-128	conv3-128	conv3-128
conv3-128	conv3-128	conv3-128
maxpool		
conv3-256	conv3-256	conv3-256
conv3-256	conv3-256	conv3-256
conv3-256	conv3-256	conv3-256
conv3-256	conv3-256	conv3-256
maxpool		
conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512
	conv3-512	conv3-512
		conv3-512
maxpool		
conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512
conv3-512	conv3-512	conv3-512
	conv3-512	conv3-512
		conv3-512
maxpool		
FC-4096		
FC-4096		
FC-1000		
softmax		

Figure 6: Architecture of VGG-21 and VGG-23 models

References

[1] Andy Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. In *International Conference on Machine Learning*, pages 1059–1071. PMLR, 2021.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[4] Wei Ma and Jun Lu. An equivalence of fully connected layer and convolutional layer. *arXiv preprint arXiv:1712.01252*, 2017.

[5] Stéphane Mallat. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150203, 2016.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[7] Andrea Vedaldi and Andrew Zisserman. Vgg convolutional neural networks practical. *Department of Engineering Science, University of Oxford*, 66, 2016.

[8] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.