Resolution cascade for a reduction of the computational cost

Rapport 1

Ilyas Cherifi Alaoui

Research student
Department of Software Engineering and Information Technology
Efficient image classification
ETS
ilyas.cherifialaoui@etu.emse.fr

Programme : Stage Mitacs ETS
Internship tutor : Marco Pedersoli
Internship supervisor : Sonia Salimani

Résumé

In this initial report, I discuss the work I have conducted under the guidance of Mr. Pedersoli at ETS Montreal. The focus of my research revolves around the performance of image classification networks. To initiate my study, I begin with a fundamental observation : the performance of an artificial network inherently relies on the resolution of the images it processes. For instance, the complexity of learning and predicting using a fully connected network increases proportionally with the number of pixels in the input images. In the case of networks featuring convolutional layers, we can anticipate a similar dependence. Therefore, if we aim to strike a balance between classification accuracy and network complexity, it becomes imperative to make a thoughtful selection regarding the resolution of the utilized images.

The concept that I have implemented involves employing multiple networks, all trained with the same set of images but with varying resolutions. When making predictions for a given image, our approach involves initially attempting classification at a lower resolution. If the prediction does not meet the required level of accuracy, we then progressively increase the resolution until we reach the original resolution.

Table des matières

Table des matières

Chapitre 1

Historic

1.1 Input Size and CNN Classifier

1.1.1 Fully Convolutional Neural Networks

Richter et al. [7] have studied the influence of input size on CNN networks. They have shown that bigger images do not necessarily yield better results. Instead, there exists an optimal size for each model.

Richter et al.'s work leads us to the conclusion that models do not perform worse on smaller images solely due to reduced information but also because the model needs to be adapted to input sizes. For cascade architectures, models should be well-suited to each size.

1.1.2 Effects of Varying Image Quality on Image Classification Performance

Simple visual information does not vary significantly, but complex visual information changes drastically with the reduction of image resolution (Suresh Prasad Kannojia et al. [4]). This assumption has motivated our choice to use high resolution only when necessary.

Samuel Dodge et al. [2] degraded image quality by introducing blur, noise, or using compression such as JPEG and studied its impact.

M. Chevalier et al. [1] worked on fine-grained classification problems and demonstrated that above a certain threshold (beyond a reduction of one half), the model's accuracy remains stable.

In all of these studies, we realize that beyond a certain reduction factor, precision changes very little. Therefore, in our quest to find a compromise between precision and performance, this result is noteworthy.

1.1.3 EfficientNet : Research on Adjusting Resolution, Width, and Height of Models

Mingxing Tan et al. [8] were the first to empirically quantify the relationship among all three dimensions of a network : width, depth, and resolution. As we intend to use different models for different resolutions, in order to achieve the best possible results, we need to adapt the width and depth to the resolutions we use.

For our work, we will use models from ResNet. We can choose the depth of ResNet; there are five versions of ResNet with 18, 34, 50, 101, or 152 layers.

MobileNet [3] can also be of interest because the width of the model can be adjusted.

1.2 Multi-Scale Architecture

The multi-scale architecture is a natural response to the challenges posed by datasets with varying levels of intricacy, granularity, and context. Multi-scale architectures are mainly used to enable the extraction of relevant features at different levels of detail. We have identified some relevant work on multi-scale architectures.

1.2.1 Convolutional Neural Network Cascade

In the publication by Haoxiang Li et al. [5], a cascade architecture is described. The goal is to improve the performance of a facial detection model. Haoxiang Li et al. aimed to address the challenges posed by the large visual variations of human faces in cluttered backgrounds and the extensive search space of possible face positions and sizes.

The proposed CNN cascade operates at multiple resolutions, quickly rejecting background regions in the faster, lower-resolution stages. The benefits of this architecture include enabling fast evaluation and the rapid early rejection of false positive detections.

1.2.2 Pyramid Architecture

Pyramid architectures are very popular in object detection and are often used to increase accuracy at the expense of computation time. Researchers like Tsung-Yi Lin et al. [6] have developed feature pyramids with minimal additional computational costs.

These pyramid architectures can, for example, use different fusion feature maps derived from various convolutions and combine the results to improve performance. In our case, we can draw parallels to a pyramid architecture, moving from the top of the pyramid to the bottom. The connections between the levels are established based on the precision of predictions from the previous levels.

1.3 Conclusion

Downscaling the input images of a model will reduce fine-grained details. If the model is well adapted to the size, we may experience a loss in performance. However, it provides a different perspective on images and can result in different evaluations of false positive frames.

Enabling detection across a range of scales has shown to be beneficial in pyramidal architectures. In our case, we are willing to skip certain pyramid levels under specific conditions. Thus, we have interesting parameters to work with to improve inference time.

Chapitre 2

Cascade architecture

2.1 Structure from low to high resolution

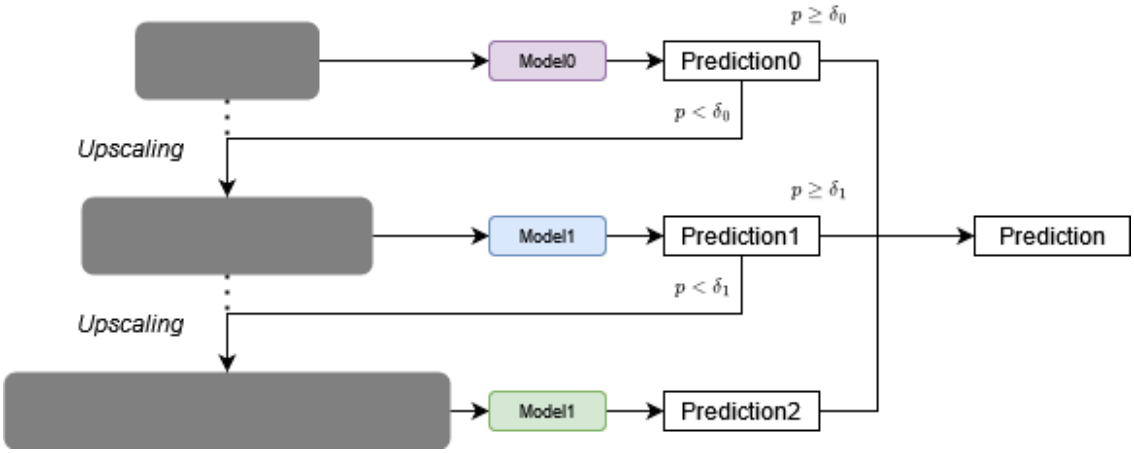

FIGURE 2.1 – Scheme of the cascade

2.1.1 Mathematique description

Our architecture consists of a series of n models

$$M = \{m_0, m_1, ..., m_n\}$$

For each model corresponds a scale factor and an inputs set :

$$F = \{f_0, f_1, ..., f_n\}$$

Considering that : $0 < f_0 < f_1 < ... < f_n < 1$

$$V_I = \{I_0, I_1, ..., I_n\}$$

$$for\ p \in [0...n], I_p = \{i_0, i_1, ..., i_{n_p}\}$$

In this architecture, I_p is defined recursively. Let ϕ be the prediction function and δ_p the precision threshold used on m_p. We point out that for an image i_p, $\phi(i_p)$ is a list with values between 0 and 1 for each predicted class :

$$p = 0,\ I_0 = Images\ to\ predict$$
$$p > 0,\ I_p = \{i \in I_{p-1} \mid \forall\, p \in \phi(i),\ p < \delta_p\} \tag{2.1}$$

If no class have a prediction probability high enough (all p < δ_p), the image will be part of the next model's prediction to try to to achieve a better prediction.

2.1.2 Evaluation of Complexity

To measure the computing time, we count the FLOPs (Floating Point Operations) of the cascade architecture.

$FLOP_p$ with $p \in \{0, 1, 2\}$ represents the number of FLOPs for each model used with a 224x224 input resolution.

We can then calculate average FLOP of the entire the architecture :

$$FLOP_{cc} = \sum_{p=0}^{n} FLOP_p * f_p^2 * card(I_p)$$

This is equivalent to :

$$FLOP_{cc} = FLOP_0 + \sum_{p=1}^{n} FLOP_p * f_p^2 * (1 - \sum_{k=0}^{p-1} \frac{card(I_p) - card(I_{p+1})}{card(I_0)})$$

We introduce $x_p = \frac{card(I_p) - card(I_{p+1})}{card(I_0)}$ which represents the proportion of image classified by the model m_p.

$$FLOP_{cc} = FLOP_0 + \sum_{p=1}^{n} FLOP_p * f_p^2 * (1 - \sum_{k=0}^{p-1} x_p) \tag{2.2}$$

In the remainder of our study, the term "FLOP" will refer to this calculation. This represents an average FLOP per image, which intrinsically depends on the data and models used.

2.2 Experiment

2.2.1 Dataset

We are using Imagenette-320, a subset of Imagenet with 10 classes, at resolutions of 32, 96, and 224.

2.2.2 Model Choices

Using Well-Known Models

Since the dataset is relatively small (9469 images for training), we have opted for ResNet18, ResNet34, and ResNet50.

Our objective is to demonstrate that we can reduce the number of FLOPs for ResNet34 while maintaining the same level of accuracy. Additionally, we aim to significantly reduce the number of FLOPs for ResNet50 while achieving a performance close to the original.

Depth and Performance

Studies have shown that the depth of a model impacts its performance at different resolutions. As a result, we are using ResNet18 at a resolution of 32x32, ResNet34 at 96x96, and ResNet50 at 224x224. The performance of these models is summarized in Table 2.1.

Furthermore, opting for networks with smaller depths will inherently reduce the number of FLOPs.

Models	Resolution	$FLOPS_{224}$(M)	$FLOPS_{Res}$(M)	$acc@1_{res}$	$acc@1_{224}$
ResNet18	32	11.69	0,24	72.66	88.20
ResNet34	96	21.80	4,00	85.40	88.92
ResNet50	224	25.56	25.56	89.63	89.55

TABLE 2.1 – Performance of model used for cascade

2.2.3 Choice of threshold

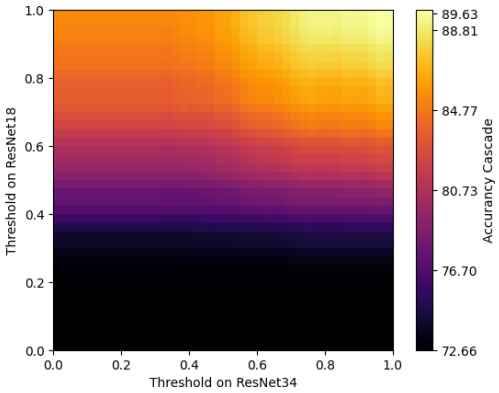

FIGURE 2.2 – Map of Cascade Accuracy on Imagenette

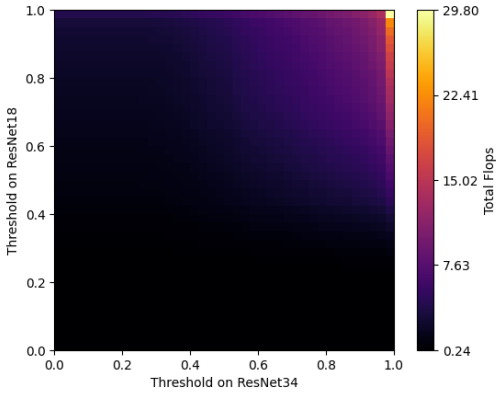

FIGURE 2.3 – Map of FLOPs on Imagenette

The cascade thresholds we have introduced are powerful hyperparameters that allow us to achieve the ideal balance between performance and computational cost.

To determine the ideal threshold, we have created maps showing the accuracy and FLOPs of the cascades based on the chosen thresholds.

In Figure 2.2, we can observe that the accuracy approaches closer to 1 as the threshold value increases.

For the various thresholds, we can calculate the number of FLOPs using Equation (2.2) and then present the results in Figure 2.6. To enhance readability, we have truncated the FLOP data for the last layer of ResNet34.

2.2.4 Exploiting Results

In Figure 2.6, for each targeted accuracy, we have identified several pairs of thresholds that achieve an accuracy within approximately 0.01% of the targeted accuracy.Then

we took the couple with the best resulting FLOPs.

2.2.5 Choice of threshold

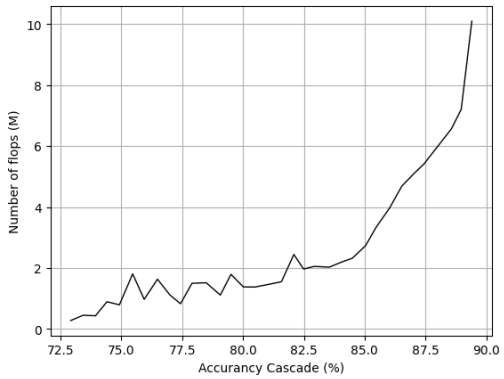

FIGURE 2.4 – FLOPs as a function of acc. Work domain : threshold from 0,0 to 1,1 (40x40 steps). Accuracy precision : 0.1% Max accuracy : 89.40%

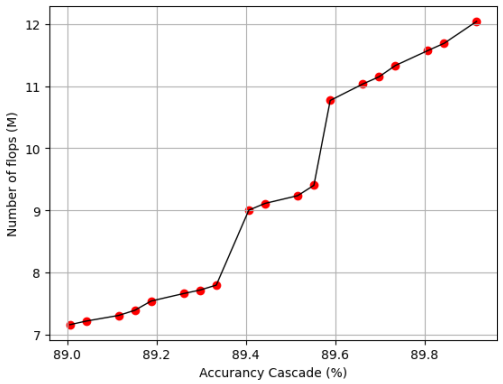

FIGURE 2.5 – FLOPs as a function of acc. Work domain : threshold from 0.97,0.74 to 1,1 (50x100 steps). Accuracy precision : 0.02% Max accuracy : 89.91%

If we search with more precision we can find better results. And we can find threshold that lead to better performance than the top model.

Improving the medium model

Lets assume we are taking ResNet34 as reference. Assuming that the objectif is to frame a model between a "big" and a "small" model to tune it performance and efficiency. For a resolution of 224x224 ResNet34 has a accuracy of 88.92 and a number of flops equal to 25.56 M, our architecture cam reach 89.0 with 7.15 M. Which is 3.0x less.

2.2.6 Improving the efficiency of the higher model

Lets assume we are taking ResNet152 as reference. If the objective is to lower the the FLOPs of the top model.

For the accuracy 89.66, our architecture succeed to have 11.04 M FLOPs. Which is 2.3x less FLOPS for an equivalent architecture.

By scarifying between 1 to 0.1 % we can have we can win from to 4.4x to 3.2x the number of FLOPS. 2.7

2.2.7 Test on test data

TValidation was utilized to determine the best threshold for each targeted accuracy. Subsequently, we applied these selected thresholds to the test data to assess their performance. We generated a similar graph based on the test results.

On the test subset, the models exhibited slight variations in performance : ResNet18 achieved an accuracy of 71.65%, ResNet34 achieved 85.82%, and ResNet50 achieved 89.47%. These differences in accuracy between the models may account for some of the disparities observed between the validation graph and the test results.

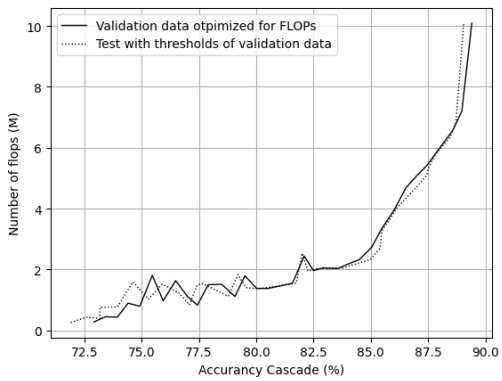

FIGURE 2.6 – FLOPs as a function of acc. Taking the threshold from validation and use it on test. Test Max accuracy : 89.05%

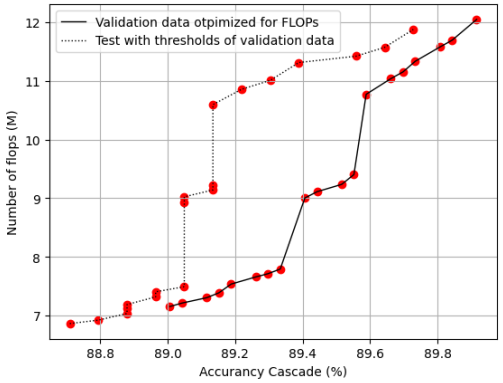

FIGURE 2.7 – FLOPs as a function of acc. Taking the threshold from validation and use it on test. Test Max accuracy : 89.73 %

Conclusion

Our work has demonstrated that a resolution cascade architecture can serve as a tool to strike a balance between performance and complexity in the context of image classification.

Bibliographie

[1] Marion Chevalier, Nicolas Thome, Matthieu Cord, Jérôme Fournier, Gilles Henaff, and Elodie Dusch. Lr-cnn for fine-grained classification with varying resolution. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3101–3105. IEEE, 2015.

[2] Samuel Dodge and Lina Karam. Understanding how image quality affects deep neural networks. In *2016 eighth international conference on quality of multimedia experience (QoMEX)*, pages 1–6. IEEE, 2016.

[3] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets : Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv :1704.04861*, 2017.

[4] Suresh Prasad Kannojia and Gaurav Jaiswal. Effects of varying resolution on performance of cnn based image classification : An experimental study. *Int. J. Comput. Sci. Eng*, 6(9) :451–456, 2018.

[5] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. A convolutional neural network cascade for face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[6] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[7] Mats L. Richter, Wolf Byttner, Ulf Krumnack, Anna Wiedenroth, Ludwig Schallner, and Justin Shenk. (input) size matters for cnn classifiers. In Igor Farkaš, Paolo Masulli, Sebastian Otte, and Stefan Wermter, editors, *Artificial Neural Networks and Machine Learning – ICANN 2021*, pages 133–144, Cham, 2021. Springer International Publishing.

[8] Mingxing Tan and Quoc Le. EfficientNet : Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019.