Convolutional neural network for image classification

Saul Gómez
Universidad de Los Andes
sc.gomez11@uniandes.edu.co

Diego Valderrama
Universidad de Los Andes
df.valderrama@uniandes.edu.co

Abstract

Convolutional Neural Networks (CNN) have become the most widely used system to address the image classification task today. Due to their powerful ability to learn patterns have acquired great importance for the realization of a lot of problems in the field of computer vision. Because of this, a CNN was implemented with four convolutional layers and two fully connected layers, applying max pooling on the first layer and a nonlinear activation function on the final layer output. The model was trained with 60.000 images of the database training set, 100 images per batch and 100 epochs from the Large-scale CelebFaces Attributes (CelebA) Dataset. The performance obtained was 66.8% with precision and recall of 0.878 and 0.539, respectively.

1. Introduction

Image classification is defined as the task of categorizing images into one of several predefined classes and it is a fundamental problem in computer vision [9]. Likewise, it forms the basis for other computer vision tasks such as localization, detection, and segmentation [2]. Even though the action of discriminating objects is a natural activity develop by humans, it represents a challenging task for computer due to some complications encountered in the representation of real world by digital images such as viewpoint-dependent object variability and the high in-class variability of having many object types [1]. However, deep learning models that exploit multiple layers of nonlinear information processing, for feature extraction for pattern analysis and classification, have been shown to overcome these challenges and exposed promising results [9].

One of the most implemented architectures to carry out an image recognition task corresponds to the convolutional neural networks (CNN). Neural networks are computational techniques for recognizing patterns, which consists of simple processing units, called neurons or nodes, connected to other neurons in the network by unidirectional connections of different strength or weight [8]. Usually, the neurons are arranged in a series of layers, bounded by input and output layers encompassing a variable number of hidden layers, connected in a structure according to the complexity of the problem addressed [8]. The most common architecture of a standard feedforward CNN consists of a convolutional and pooling layers grouped into modules and one or more fully connected layer [9].

The convolutional layers serve as feature extractors, and thus they learn the feature representations of their input images. The neurons in the convolutional layers are arranged into feature maps and inputs are convolved with the learned weights in order to compute a new feature map, and the convolved results are sent through a nonlinear activation function [3]. On the other hand, the purpose of the pooling layers is to reduce the spatial resolution of the feature maps and achieving spatial invariance to input distortions and translations. Finally, a layer is fully connected if each node in the layer is connected to all the nodes in an adjacent layer [5], [4].

CNNs are feedforward networks, that is information flow takes place in one direction only, from their inputs to their outputs [9]. In addition, two of the main particular characteristics of these models is that the input is the image without any processing and that the representation space and the classifier used are the network itself. On the other hand, these models implement learning algorithms during training stage to adjust their parameters such as weights and biases in order to obtain a desired output. Therefore, an image is input directly to the network, and this is followed by several stages of convolution and pooling. Later, representations from these operations feed one or more fully connected layers and the last fully connected layer outputs the class label [9]. Finally, backpropagation algorithm is employed to compute the gradient of an objective to determine an adjustment factor to update the parameters in order to minimize errors that affects performance [6].

Keeping this in mind, the most important aspect of networks are their ability to learn from examples, and because the learnt information is stored across the network weights, to generalize, achieving accurate classifications even for input patterns not included in the training set [8]. In this order of ideas, a convolutional neural network was implemented to address a multi-class and multi-label classification problem of celebrities with different attributes.

2. Methods

2.1. Database

Large-scale CelebFaces Attributes (CelebA) Dataset was used. It consists of 200.000 images of celebrities and it is labeled with 40 different attributes (not-mutually exclusive) such as wearing eyeglasses, smiling, arched eyebrows, male/female (do not judge about binary genders), pale skin, young/old. The images in the dataset cover large pose variations and background clutter [7]. For this study, only 10 out 40 attributes were used to train the model. Figure 1 shows examples of the images employed for the training stage.

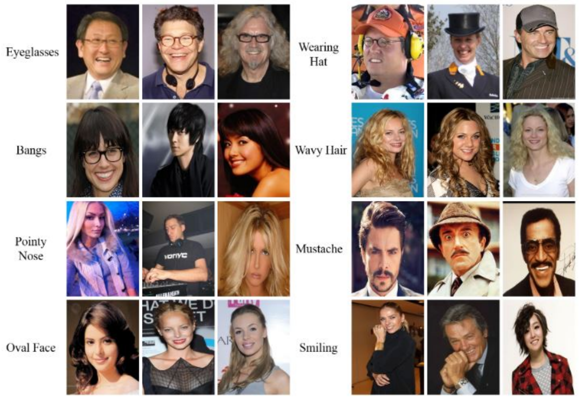

Figure 1. Sample images of different classes among the whole dataset

2.2. Algorithm

A CNN was implemented with four convolutional layers and two fully connected layers. In the first convolutional layer max pooling was performed, and it was applied down-sampling in the first two layers and up sampling in the last two. Each batch was normalized. Additionally, a dropout was applied to convolutional layers 2, 3 and 4. Likewise, in the layer where max pooling is passed from down-sampling to up-sampling a greater dropout than in the other layers was applied in order to avoid the overfitting of the model. Finally, since it is a multilabel classification problem, non-linear layers were not used within the network architecture but a sigmoid activation function was used to normalize the predictions at the output of the last layer fully Connected. The model was trained with 60.000 images from the train set divided in batches of 100 images and 100 epochs. In addition, ablation studies were carried out to explain the functioning of the developed network, removing the third convolutional layer and the first fully connected layer.

3. Results

The results were obtained from the calculation of the F-measure by the accuracy and recall of the model for the prediction of the labels in each of the images of the test set. In table 3 performance results of the implemented model for the celebA database are depicted. Performance was 66.8%, with 88% and 54% precision and recall, respectively.

Class	Accuracy (TP)	Precision	Recall	F-measure
Eyeglasses	1153/1289	0.991	0.894	0.940
Bangs	1812/3109	0.972	0.583	0.729
Black Hair	2327/5422	0.825	0.429	0.564
Blond Hair	504/2660	0.980	0.189	0.317
Brown Hair	327/3587	0.707	0.091	0.161
Gray Hair	115/636	0.975	0.182	0.307
Male	7073/7715	0.932	0.917	0.924
Pale Skin	204/840	0.933	0.243	0.386
Smiling	8804/9987	0.893	0.882	0.887
Young	14843/15114	0.571	0.982	0.722
AVG.	53.9%	0.878	0.539	0.668

Additionally, the table 3 shows the results obtained for the performance of the convolutional neural network implemented after removing the third convolutional layer and the first fully Connected layer. As can be notice, the performance decreased around 4 and 2 percentage points, respectively

Model	Precision	Recall	F-measure
Original	0.878	0.539	0.668
No convolutional layer 3	0.823	0.502	0.624
No convolutional layer 3 nor fully conected	0.830	0.523	0.642

Finally, in figure 2 it is shown the performance of the implemented model in the classification task for sample images of the test set. For each image a vector of zeros and ones is depicted which one correspond to an attributed recognized in the image and zero means that the attribute is not present. The attributes Eyeglasses, Bangs, Black Hair, Blond Hair, Brown Hair, Gray Hair, Male, Pale Skin, Smiling and Young refers to the same order positions of the vector, respectively.

4. Discussion

Alternatively to the implemented model, another mechanism to address this problem is by implementing a model with Bag of Words (BoW), whose main objective is to design visual words and to represent the image by its distribution of visual words. First of all, the local features of

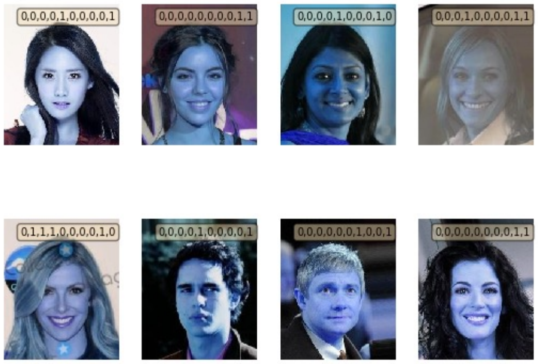

Figure 2. Qualitative performance of the model implemented in the multi-label classification of images of the test set. Labels are Eyeglasses, Bangs, Black Hair, Blond Hair, Brown Hair, Gray Hair, Male, Pale Skin, Smiling and Young refers to the same order positions of the vector, respectively.

the image are extracted by sweeping with a pixel to pixel slider by computing Sift dense across the whole set of training images. At the same time, this process is carried out at the different levels of the Gaussian pyramid with the aim of making the model invariant at the scale. Once the Features are extracted, the dictionary of visual words is constructed by clustering with k-Means in the Sift space. Next, the pyramid of visual word histograms is computed for each image and the corresponding classifier is trained, in this case Random Forest could be a good alternative. Finally, to evaluate the performance of the model the pyramid of visual word histograms is computed for each test image and evaluated with the classifier, assigning the label of the class with a higher level of confidence.

Focusing on the model implemented, according to the results presented in the table 3, the model implemented presents an adequate precision for the classification of these data, however, the coverage barely exceeds half of the instances to be evaluated, which indicates that the model is ignoring much of the information to perform the classification. This can be explained by the fact that not all training images were used to train the model and that the network is not deep enough to properly learn the parameters that decrease the error in the classification of each class. On the other hand, there is a pattern among the classes that represent the greatest difficulty for the model, which are blond hair, brown hair, gray hair and pale skin. The pattern identified refers to the importance of color intensities as characteristic features of these classes. So, when considering grayscale images, the weight of the intensities is reduced to a single channel, which makes it difficult to discriminate this type of categories. However, it should be noted that the performance obtained by the implemented model is significantly high considering the shallowness of the architecture.

The challenges faced in the design of the model's architecture are summarized in two main factors. The first is due to the adjustment of the dimensions at the outputs of each of the convolutional layers and the entry of the next layer, so that the sizes of the image display match and are appropriate to extract the necessary characteristics of the objects for correct network learning. The second challenge concerns the design of a sufficiently deep architecture taking into account the limited resources available, so that the amount of layers was appropriate for learning the main characteristics of each of the classes to be evaluated.On the other hand, compared to the original design, a convolutional layer was added with respect to the three initials of the network implemented in the previous challenge and it was decided to use a detector of faces as input of the neural network, so that it forces the network to learn patterns related to the expressions and attributes of the person ignoring as much as possible the background noise of the image and artifacts that are not points of interest and do not contribute to the learning.

Now, according to the results observed in the table 3, this in principle explains the importance of this convolutional layer within the network architecture, since the performance of the model is reduced by up to 4%. However, by also removing the first fully connected layer, having a direct classification from the last convolutional layer output to the 10 classes, the performance increased again. This means that there is a relationship between the removed convolutional layer and the overfitting of the model to the data. However, if you also remove the first fully connected layer, the network is able to generalize acceptably again, which is reflected in increased performance. Also, the relevance of this layer in architecture may be due to the fact that it is the point where it passes from applying down sampling to up Sampling, allowing a better extraction of the low level features of the image and therefore a better generalization.

Finally, according to the results presented in the figure 2 and taking into account the previously mentioned difficult classes, it is possible to appreciate the lack of precision by the model for the attribute blond hair, for example. Here are three examples of people with blond hair, to whom no such attribute was assigned (See figure 2 images 4, 5 and 7 from left-right top-down).

5. Conclusions

Although the architecture used for the implemented model is shallow, the network manages to learn characteristic patterns of each of the classes, which is a good first approach to address the problem. However, it is advisable to increase the number of convolutional layers in order to increase the depth of the network and to induce a learning that improves generalization. Also, taking into account that the main defect of the model is a deficiency in recall, it is necessary to increase the number of images used for model training, as well as implementing an alternative to balance the batch. At the same time, it may be useful to augment the data, mainly for the classes that represent the greatest difficulty to the model, in order to improve the learning of patterns.

References

[1] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *IJCAI*, 2011.

[2] A. Karpathy et al. Cs231n convolutional neural networks for visual recognition. *Neural networks*, 1, 2016.

[3] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[4] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[5] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[6] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[7] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[8] A. S. Miller, B. H. Blott, and T. Hames. Review of neural network applications in medical imaging and signal processing. *Medical and Biological Engineering and Computing*, 30:449–464, 1992.

[9] W. Rawat and Z. Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29:2352–2449, 2017.