RIGHT CONVOLUTIONAL NEURAL NETWORK FOR CLASSIFICATION ILLUSTRATIONS IN ARTWORKS

Pavel Sikora
Doctoral Degree Programme (third year), FEEC BUT
E-mail: xsikor14@stud.feec.vutbr.cz

Supervised by: Kamil Riha
E-mail: rihak@feec.vutbr.cz

Abstract: This paper deals with the image classification problem in the field of artworks. The article uses a custom dataset from artworks with eight classes of some not common objects and illustrations. This dataset is used to train three convolutional neural networks for classification. All classification results are well discussed and evaluated with an example on the images from a dataset.

Keywords: artwork, convolutional neural network, deep learning, image classification, keras, machine learning

1 INTRODUCTION

Nowadays in the world is experiencing huge benefits of neural networks, fall into today's so-called category AI (Artificial Intelligence). Since 2014 very deep convolutional networks have been used more widely [1]. Nowadays AI is used in many industries from medicine to automatic robots in halls to automatizing tasks in mobile phones, which is used by the vast majority of people. Thanks to the rapid trend of increasing the performance of computer technology and GPUs, researchers can create more complex and deeper networks. Thanks to that improvements, AI technology is more accurate and can be used in the next industries. There are also uncharted ends of this technology, one of them is works of art. [1, 2, 3]

This paper focuses on artworks, where can be applied the classification technique. The main interest is objects or illustrations that do not occur in commonly available datasets. For that, is used custom dataset which is made from artworks created by the Vasulka's couple [4].

2 METHODOLOGY

Today's classification tasks greatly solving CNN (Convolutional Neural Network). As the name suggests, CNN is DNN (Deep Neural Network) which among others uses convolutional layers. Nowadays exist a lot of image classification DNN models. In the end, three convolutional networks were chosen, namely VGG [3], DenseNet (Dense convolutional Network) [2] and Inception [1]. All these models handle very well with image classification problematics.

A F1-score is used to evaluate all results. This metric is calculated from precision and recall. The calculation formula is [5]:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{1}$$

2.1 DATASET PREPARATION

This article focuses on the artwork dataset contains these objects and illustrations: **air, fire, image processing keying, machine vision (fisheye), woody, stripes, interior, landscape**. The dataset

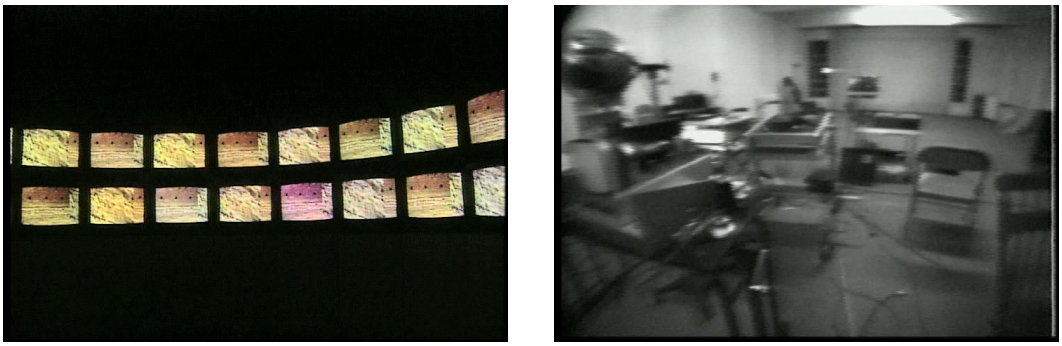

Figure 1: Two images from a dataset [4]

contains images with the sky, for which is a dedicated class *air*. Artworks sometimes contain scenes depicting the fire, for that is class *fire*. *Image processing keying* is on images where technique keying is applied. *Machine vision (fisheye)* occurred in images that are recorded uses the fisheye lens. Class *woody* suggests images where is Woody, one of the images dataset authors. *Stripes* represent specific lines. An example of *stripes* is on the left side of the figure 1. The second image in figure 1 shows the class *interior*, which classifies images that are captured in interiors. Images where occur landscapes are classified with *landscape* class.

At the beginning of the whole process, the dataset must be prepared into a suitable form for training. Dataset has been separated into folders, where folder names are class names that occur in images in that folder. Although the model is trained with a multi-label technique. Thanks to this technique trained model can classify all trained classes in one image, not only one class per image. This hierarchy is loaded to memory, shuffled, and saved to *CSV file*. Next, this file is used to create a data generator in Keras.

All used CNNs have been modified to train for eight classes. For that classification parts of every CNN model were replaced. Firstly with 2D global average pooling and dropout layer. Dropout layer use 40 % rate. Directly behind these layers, eight dense layers were added. Every dense layer is responsible for evaluating one of the eight classes [6]. All these dense layers use the sigmoid activation function. Binary cross-entropy is used to compute the loss function. The training process uses optimizer Adam with learning rate 1^{-5}, with adjustment during training to a minimum 1^{-6}.

2.2 VGG

The network input is an RGB image with a size of 224×224 pixels. The convolution layers used in the VGG model use very small receptive fields of size 3×3 with step one. The use of such small convolution filters allows the VGG architecture to use a larger number of layers, as the authors have researched [3]. They also use convolutional filters of size 1×1, which are used to create a linear transformation of the input, followed by the use of ReLU (Rectified Linear Unit). These filters are used to create a crucial function more non-linearly without changing the receptive fields. [3, 7]

In this paper is used VGG 19, which consist of most layers and could be most accurate.

2.3 DenseNet

DenseNet comes with direct connections between any layers consists of the same feature-map size, and as the name suggests, the model uses a dense connectivity pattern. This model naturally scales to hundreds of layers without any difficulties in optimization. DenseNet is distinguished by less parameters and computational complexity to achieve great performance. The input of the network is an RGB image with size 224×224 pixels. This work uses DenseNet 201. [2]

2.4 INCEPTION

The inception model passed two improvements. This paper uses the third version of that model. The authors claim that the model is computationally efficient. Inception V3 works with factorizing convolutions into smaller convolutions, spatial factorizing, auxiliary classifiers, and efficient grid size reduction. The network input is an RGB image with a size of 229×229 pixels. [1]

3 EXPERIMENT AND RESULTS

Training environment consists of Keras with TensorFlow backend [8]. The whole environment uses a middle-class workstation but with GPU Nvidia 2080 Ti for accelerating all training processes. For this paper, F1-Score is computed with scikit learn library [5]. The custom dataset is not balanced. Because of this fact, we used a technique using class weight. For each class, a weight is calculated at the beginning of the training. Weights prevent the degradation of classes with low image representation.

Classification example of one model can be seen in figure 2.

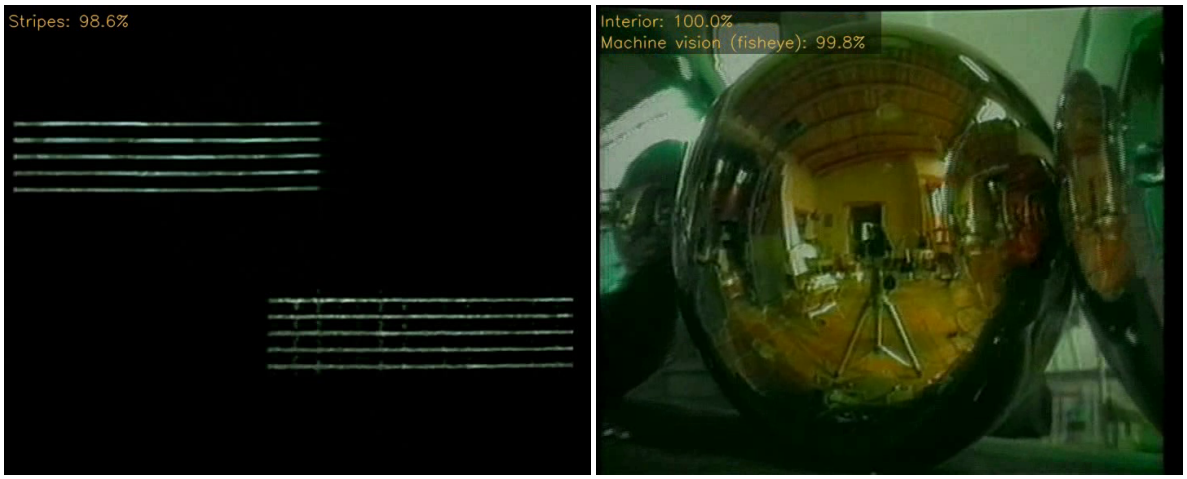

Figure 2: Two classified images

On the left image is an example of stripes, where the model great classify stripes with 98,6 % probability. The image on the right side of the figure shows a mirror ball in the interior. Model right classifies interior with great probability 100 %, but the model also classifies fisheye lens, which could be not exact. Probability is 99,8 %. When is it compared with images captured with a fisheye lens, images are very similar. Images have also a warped scene, but the fisheye lens warps the whole image, not the only center of the image as shows this image.

Table 1: Average F1-score for all classes per model

Model	Average [%]
VGG	*69,4*
DenseNet	**79,9**
Inception	74,7

Table 2: Classes statistics

Class name	Average F1 [%]	Number of images
Air	72,8	3048
Fire	58,4	*789*
Image processing keying	60,0	3595
Interior	88,3	**10561**
Landscape	*57,7*	4921
Machine vision (fisheye)	**88,9**	1834
Stripes	83,3	2057
Woody	87,8	3161

This experiment shows that the best model for classifying illustrations in artworks is DenseNet 201, which has the best average F1-score 79,9 % of tested models. Average F1-scores are in table 1. Second place has Inception with a score of 5,2 % lower than the best model. The worst model is VGG which gets a score of 10,5 % lower than the best model.

Best average F1-score get class Machine vision (fisheye), which get 88,9 %, as shows table 2. The worst classified class in order of average F1-score is a landscape with 57,7 %. As mentioned above, the dataset is not balanced, as can be seen in table 2. With the class balance technique, this fact is not so significant. Most images, specifically 10561, contain class interior. On the other hand, at least images contain the class fire, with 789 images.

In a deeper look into the classification of particular classes, also DenseNet get highest F1-score. Model get 94,5 % in classification *woody* class, as shows figure 3. Inception model get second place as in average score, and also *woody* class get best value, which is 93,4 %. VGG classify *woody* class as second best in order of F1-score with 75,4 %. Best classified class with VGG model is *interior*, which get 89,2 %. *Interior* class is on third place classifying by model DenseNet with 89,7 % and fourth place classifying by model Inception with 86,1 %.

VGG and DenseNet classify the worst *landscape* class, with 46 %, respectively 63,2 %. Inception classify the worst *fire* with 49 %. From these results, can be said, that *woody* class is the most accurate classifiable. But with regard to the average F1-score of classes, Machine vision (fisheye) get 88,9 %, which is best. The worst classifiable class is fire as is the same according to the average F1-score of classes.

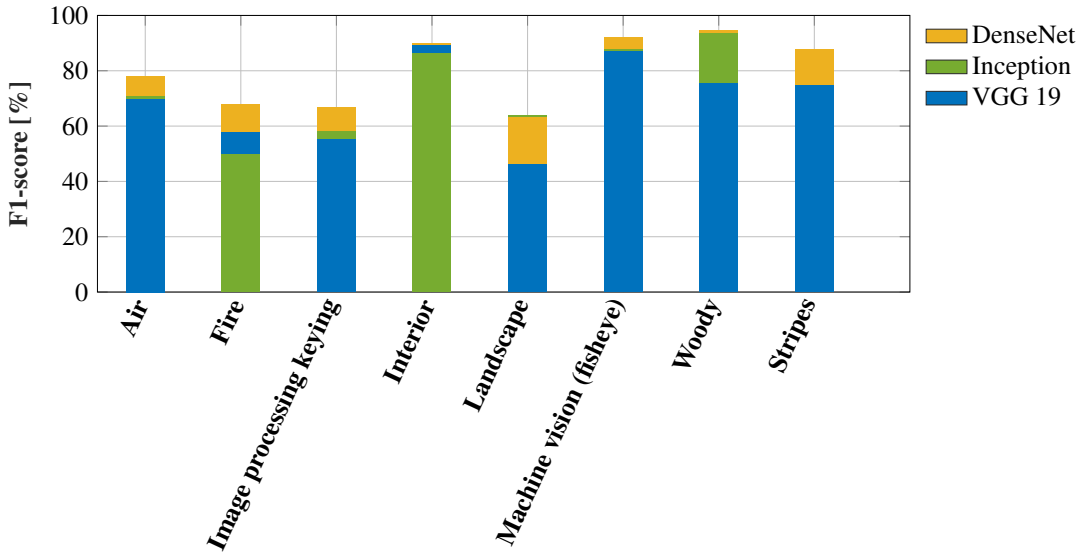

Figure 3: F1-scores of every model for every class

The Inception V3 model is very robust and has very good accuracy. Comparing these three models on ImageNet dataset [9], different order will get. Best is the Inception model with TOP-1 accuracy 78,8 %, but on the artwork dataset, the Inception model gets second place. On ImageNet, the second is DenseNet 201 with 77,42 % TOP-1 accuracy, but in the artwork dataset, DenseNet took first place. The third place is the same and is taken by VGG 19 with 74,5 % TOP1-accuracy on ImageNet. [10]

4 CONCLUSION

This paper proposed right deep CNN for classification artworks, which is DenseNet 201 as is shown on test results. This model get 79,9 % F1-score. At all three CNN models has been trained for classification custom artwork dataset. Article not used common dataset, but prepared custom dataset from artworks with specific classes to classify. The article also finds out that although works of art contain materials similar to those found in common datasets, the resulting accuracy may vary and it may not be sufficient to compare models only on general datasets.

ACKNOWLEDGEMENT

This work was supported by the Technology Agency of the Czech Republic under project TL02000270 Media Art Live Archive: Intelligent Interface for Interactive Mediation of Cultural Heritage.

REFERENCES

[1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[2] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[4] W. Vasulka and V. S., "Binary lives," *Source: Archiv Vasulka Kitchen Brno*, 1999.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[6] V. J, "Multi-label image classification tutorial with keras imagedatagenerator," Mar 2020. [Online]. Available: https://vijayabhaskar96.medium.com/multi-label-image-classification-tutorial-with-keras-imagedatagenerator-cd541f8eaf24

[7] J. Wei, "Vgg neural networks: The next step after alexnet," Jul 2019. [Online]. Available: https://towardsdatascience.com/vgg-neural-networks-the-next-step-after-alexnet-3f91fa9ffe2c

[8] "Tensorflow." [Online]. Available: https://www.tensorflow.org/

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[10] "Papers with code - imagenet benchmark (image classification)." [Online]. Available: https://paperswithcode.com/sota/image-classification-on-imagenet