DEEP LEARNING PROJECT

CLASSIFICATION OF IMAGES WITH CNN AND AN IMPLEMENTATION OF A GAN TO GENERATE NEW IMAGES

GROUP:

Alex Panchot [M20190546]

Andreia Antunes [M20190876]

Bruno Vieira [M20190922]

Leonardo Lannes [M20180036]

1. OBJECTIVE

This project was developed for the Deep Learning course of the Master's in Data Science and Advanced Analytics and consisted in developing a system that was able to classify images of interior spaces, such as restaurants, churches, gyms or waiting rooms, based on a Convolutional Neural Network. In addition, an attempt to implement a Generative Adversarial Networks has been conducted with the purpose of creating new images as an alternative to increasing the size of the dataset through traditional means.

All notebooks and related files are available on the project GitHub page on this link.

2. INTRODUCTION

Computer vision is a subfield of artificial intelligence and computer sciences that aims to capacitate computers to develop a perception of visual entities, such as images or videos. This sub-field can subsequently be divided into several tasks, such as recognition, reconstruction or modeling; nonetheless this project seldomly addresses image classification task.

Classification of images using computer vision has been widespread across different domains, such as biology, physics, robotics, architecture or health.

The main task of the project is to develop a system that is able to classify indoor scenes, built on top of a convolutional neural network. Typically, CNN classification tasks are used to classify objects instead of whole scenes. While in some cases the knowledge of a specific object will identify the scene, i.e. bowling balls and pins for a bowling alley and train cars for a subway, other scenes contain similar or identical objects. For example, books can be found in a public library, personal library, or a bookstore. Similarly, food items can be found in different types of kitchens as well as in a dining room.

To develop this classification system, a selection of images from Indoor Scene Recognition dataset was used, which comprises 67 indoor categories, with a total of 15620 images. The number of images is not the same per category. This dataset has been developed as support for the paper *"Recognizing Indoor Scenes. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)"*.

This dataset has been selected considering the additional challenge that indoor scenes represent for a classification problem. For most indoor scenes there is a wide range of both local and global

discriminative information that needs to be leveraged to solve the recognition task (Quattoni et al., 2009).

Convolutional Neural Networks, or CNNs, were employed to perform the classification of images due to its considerable success in areas such as image classification and recognition.

In addition, heatmap analysis was conducted to get a better insight on which areas of the scenes the CNN was focusing and, consequently, which specific elements were being considered by the CNN for the classification task.

System performance was evaluated based on accuracy, and close attention was paid to computational efficiency.

Upon this, an attempt to create a system capable of generating new indoor scenes was performed through a Generative Adversarial Network, or GAN. Ideally, the GAN would produce similar, yet still unique, images to the ones presented to the CNN.

Considering that the input data are real images, finding new images is not a trivial task. Data augmentation is an option, though it simply distorts existing images rather than creating completely new scenes.

There is a panoply of applications for indoor scene classification. A possible industry application would be that a robot waiter should change its behavior with respect to plates with food depending on whether the plate is still in the kitchen or at a diner's table. This task is similar to text mining where we want to gather contextual clues instead of just the "keywords" in order to better understand the sentence.

As for generating new images, an interesting use of such "fake" scenes would be for interior design. Conceivably the network could create aesthetically pleasing scenes that could help inspire real design.

3. MODELS / APPROACH

The models presented in this section were developed in Python with the aid of the package Keras. Keras is natively available in Google Colab, which is a free cloud service that supports free cloud GPU usage. Using a GPU is much faster than a CPU and this allows the testing of many different scenarios in a timely manner.

3.1. Data Preparation

The Indoor Scene Recognition dataset comprises 67 indoor categories, totaling 15620 images, with varying number of images per category. All images are in .jpeg format but with different sizes.

A selection of 25 categories was made to lower the dimensionality and improve performance, resulting in a total of 5805 images. The dataset was then split in train (60%), validation (20%) and test (20%) sets. This split facilitates training the model on a dataset and making predictions on unseen data, allowing for adjustments to prevent the model from overfitting to train data.

Data was then formatted into appropriately preprocessed floating-point tensors, making use of Keras image-processing tool ImageDataGenerator. This process included rescaling the images to 224x224 and rescaling the pixel values to the interval between 0 and 1.

At a first stage new data was generated using data augmentation technique, which consists of generating new data from training samples by means of random transformations that include shift, flip, zoom and brightness; increasing the ability for the model to generalize and improve performance.

3.2. Convolutional Neural Network

Initial approach consisted of a small CNN without regularization. While a simpler dense layer only model is a possibility, it suffers from easy overfitting and lack of generalization ability. This is because the dense layers map to the exact same location in the image. This means it cannot "see" unless something appears in the same location for every image. Thus, we used much better CNN layers for our model.

This baseline model obtained a classification accuracy of 32%.

Subsequent model experimentation was conducted using different parameter combinations and architectures. A summary of results is available on the next section of the report.

The best performing model was obtained using the pretrained network MobileNetV2. This network architecture encompasses inverted residual blocks with bottleneck features, which considerably decreases the number of features when comparing to prior versions, and it shows better performance on input images larger than 32x32.

Appendix contains the scheme of best performing model architecture, with the pretrained architecture hidden away for readability. The classification accuracy obtained was 79%.

Regarding the tradeoff between accuracy and efficiency there were no significant differences among the best network architectures that were implemented. However, it was noted that some aspects in the preprocessing of the images, such as scaling, can have major impacts in the results. In order to confirm this impact, the model that generated the most accurate results was implemented with 3 different image scales: 50x50, 150x150 and 224x224. The results were the following:

IMAGE SCALE	ACCURACY OF MODEL	TRAINING TIME (SECONDS)
50x50	0.44	99
150x150	0.75	386
224x224	0.79	746

Table 1 -Comparison of accuracy and training time for different image scales

It is noticeable that, despite being quite efficient, the first scaling option generates low accuracy. On the other hand, the second and third scaling options generate reasonable accuracy while having a considerable difference in training time, close to 100%. Given this situation it is important to analyze how relevant the difference in accuracy might be to the problem at hand since, in some situations, it might not be worth the cost. For the classification of images, in general, any difference in accuracy can be relevant.

3.3. Generative Adversarial Network

Building a GAN network is already not a trivial task for the MNIST dataset, so for these indoor images the built GAN predictably did not work immediately as intended. One of the main problems with the dataset for GANs is that the images usually do not look similar. This did not represent a problem for the CNN since it can just look for a few features for classification.
The initial GAN model was modeled off of the MNIST example that was provided in class. As explained earlier, its use of just dense layers was a limitation to its generalization ability. While it works well for the MNIST dataset, the images used require a more complicated network. After converting the generator and discriminator to CNN layers, different combinations of network sizes were tested, as well as parameters and dataset partitions. The network is basically constituted by two CNNs, with the discriminator in a similar architecture to the CNN described above. The

generator is an inverted CNN with Conv2DTranspose layers. These layers double the size of the image (upscale) each time.

The evaluation metric for the GAN can be problematic as it includes the accuracy of the discriminator. But a high accuracy does not mean that the generator is working properly. Instead, a close inspection of the images by hand may be required to determine what is called real and not.

4. RESULTS AND DISCUSSION

4.1. Convolutional Neural Network

The table below presents the setup and accuracy on test set for 3 different models.

MODEL	PRETRAINED NETWORK	PARAMETERS	DATA AUGMENTATION	ACCURACY TEST SET
Basic_CNN	N/A	Activation: Softmax Pooling: Max Pooling Pool Size: (2,2) Learning Rate: 1e-4 Epochs: 20 Batch Size: 32	N/A	0,32
CNN_Pretrained	MobileNetV2	Base Model Trainable Layers: 8 Activation: Sigmoid Pooling: Average Pool Size: (2,2) Learning Rate: 1e-4 Epochs: 20 Batch Size: 20	rotation_range = 20, zoom_range = 0.15, width_shift_range = 0.2, height_shift_range = 0.2, shear_range = 0.15, horizontal_flip = True, fill_mode = "nearest"	0,79
CNN_Pretrained	InceptionResNetV2	Base Model Trainable Layers: 8 Activation: Sigmoid Pooling: Average Pool Size: (2,2) Learning Rate: 1e-4 Epochs: 20 Batch Size: 20	rotation_range = 20, zoom_range = 0.15, width_shift_range = 0.2, height_shift_range = 0.2, shear_range = 0.15, horizontal_flip = True, fill_mode = "nearest"	0,78

Table 2 -Summary of system parameters and respective accuracy scores on test set

The classification report for the best accuracy model can be found in Table 3. As previously mentioned, this model was trained based on MobileNetV2.

The general accuracy obtained by the model was 79%, though this was not uniform among the 25 image categories. It is possible to observe that categories such as fast food restaurant, restaurant or museum have a worse classification score when compared to categories such as library, inside of bus, buffet or bowling alley. A plausible explanation might be considerable variety scenes among each category and lack of distinct features, such as book, pins or food. Even with the use of image augmentation there might be a need for more images, than the 5805 that were used to train the model, in order to obtain higher accuracy.

Another interesting point is that with the variety of categories and complexity of some images it could be difficult even for the human eye to properly classify some of the scenes.

	PRECISION	RECALL	F1-SCORE	SUPPORT
airport_inside	0,69	0,79	0,74	122
bar	0,77	0,8	0,79	121
bowling	1	0,84	0,91	43
buffet	1	0,91	0,95	22
casino	0,88	0,89	0,88	103
church_inside	0,94	0,81	0,87	36
cloister	0,86	0,79	0,83	24
concert_hall	0,93	0,67	0,78	21
elevator	1	0,85	0,92	20
fastfood_restaurant	0,58	0,48	0,52	23
florist	1	0,95	0,97	20
gameroom	0,64	0,92	0,75	25
gym	1	0,8	0,89	46
hairsalon	0,82	0,83	0,82	48
inside_bus	1	0,85	0,92	20
library	0,95	0,86	0,9	21
locker_room	0,76	0,76	0,76	50
movietheater	0,89	0,89	0,89	35
museum	0,44	0,5	0,47	34
poolinside	0,92	0,63	0,75	35
prisoncell	0,88	0,71	0,79	21
restaurant	0,66	0,76	0,71	103
subway	0,75	0,79	0,77	108
trainstation	0,67	0,73	0,7	30
waitingroom	0,84	0,7	0,76	30
accuracy			0,79	1161
macro avg	0,83	0,78	0,8	1161
weighted avg	0,8	0,79	0,79	1161

Table 3 –Classification report of best performance system on test set

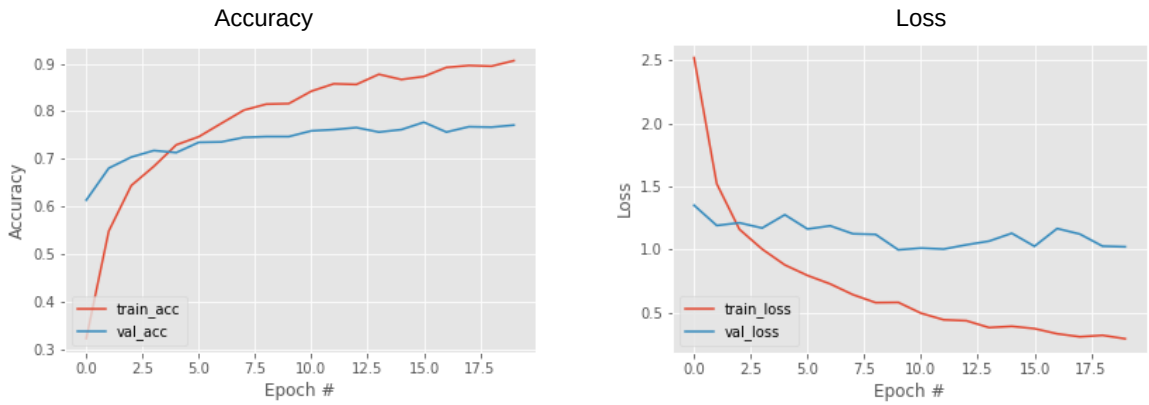


Figure 1 – Comparison of accuracy and loss scores on train and validation sets

Also, with the purpose of analyzing the quality of the model that was generated, 28 random images from the 25 categories were taken from the internet and fed to the model. Despite being a rather small data sample, this can be considered as a real-life application for the developed model. The results pointed out that the model accurately classified 17 of the 28 images, which is worse than the results obtained with the test set from the original dataset.

Figure 2 – Example of Heatmap of class activation

The heatmap images presented above are part of the new data set. The image on the left is correctly classified as airport inside, whilst the image on the right was equally classified as being of an airport inside when the reality depicts a waiting room. It is clear that the CNN was focusing on the group of people when classifying the images. However, there were images that even a human being struggle to correctly classified as belonging to a given space without being provided with proper context.

4.2. Generative Adversarial Network

However, for the GAN the images were too far apart for the discriminator to classify a created image as real. For example, the generator could create a scene that was identical to a real image but had different colored objects. This is realistic for humans, but perhaps not for the network. Using monochrome images could help remove color as a possible barrier for the GAN.

GANs also require the use of a lot of processing power. Because of this the images were scaled down to 32x32 or 64x64. This helped the network to run faster but by downscaling so much, some of the tiny details got potentially lost. This is troubling as these tiny details might be the small differences that set one scene from another. The size of the network itself is also very important. Too small a network cannot learn and the “fake” images produced are just a single color. Too large of networks take too long to learn and the “fake” images are just noise. After a long training period, the network sometimes appeared to break as well. The images produced would suddenly become just solid black. This appears to be a problem with the discriminator as the accuracy would be 100% for real and 0% for fake images. It can also be 0 and 100% respectively, but never anything besides these values.

Unfortunately, as the required processing power was too immense for even Google Colab to complete quickly, the network never completely fit any of the datasets. Large datasets are more general but require more time to fit. Smaller datasets are quicker but lead to overfitting. For all dataset sizes, the networks appeared to always be suffering from mode collapse. This means that the fake images that were produced looked nearly identical. An interesting take was that each time a set of images was produced they looked the same, but the next time a new group of images was plotted, the images were similar to each other but different from the first group.

An attempt to purposely overfit the model was conducted, in order to observe what would happen to the images and the mode collapse issue. After letting the network run for many thousands of

samples, it began to overfit and produce an image that was identical to a training sample. Unfortunately, the mode collapse problem did not go away.

Figure 3 – GAN output

From these images it is clear that the GAN was successful in overfitting this image. The image on the left was first and the one on the right was obtained after training for another 500 epochs of 4 images. On the right, it can be observed that any remaining variations between the images are gone. What is puzzling is that only one real image was being learned at a time. The next images to be plotted after the one on the right were of a different real image. After training for some time more, it was observed that the GAN also learned to copy this image as well.

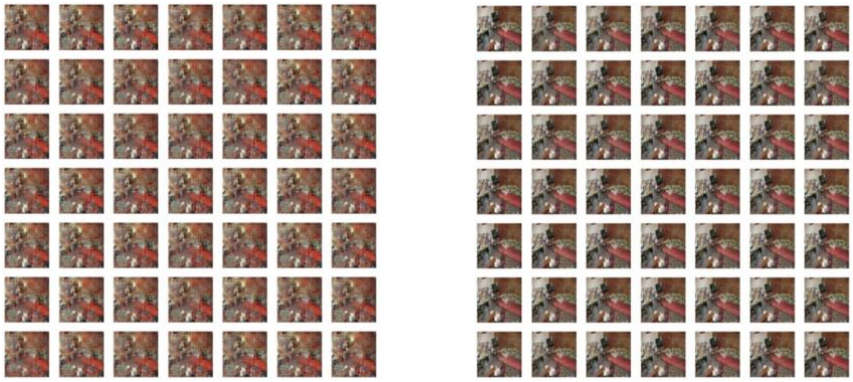


Figure 4 - GAN training with 4 images

In order to try to limit overfitting, the number of filters for each layer in the generator was reduced. However, it was clear that the network did not contain enough capacity as it kept producing the same image. The following image is not very clear but when comparing it to the previous image, one can notice that it should be the same. The lack of filters means that the network was not able to produce the fine details or gradients in color of the image.

Figure 5 – Same GAN with less filters

As the GAN was only able to overfit a couple of images over a large amount of processing time, any definite conclusion on this dataset and network can not be made, other than this GAN requires lots of power in order to test different configurations. Nonetheless, it can be assumed that the collection images are probably not sufficient on their own to create a model that is capable of generalizing.

5. CONCLUSION

Developing a classification system for indoor images has proven to be a laborious work, since the input images are very detailed and disparate. When looking at simpler datasets such as MNIST, the task of a classifying CNN or image generating GAN do not appear to be too difficult. But, when tasked with a more realistic dataset such as indoor locations, these tasks become much more difficult. In order to be able to deal with this level of detail, it is important to have a large dataset which feeds the network with enough variability of scenarios. In turn, this implies a much deeper network, which requires more time and computational power to train.

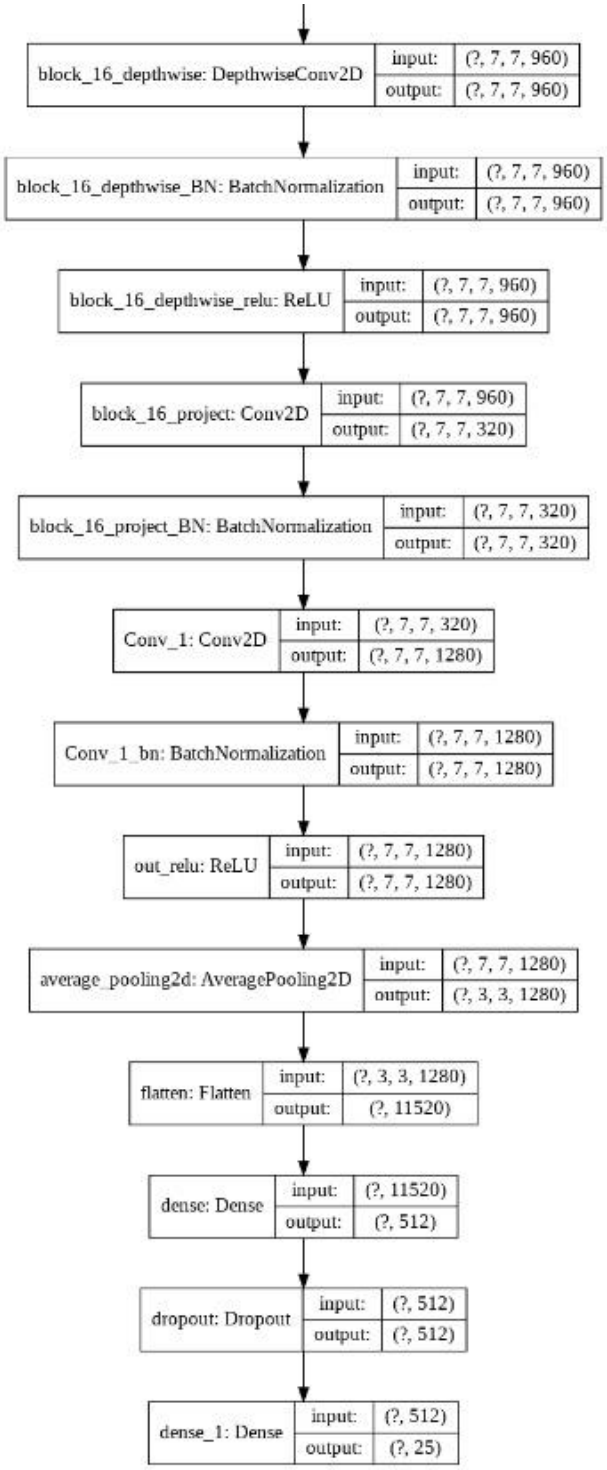
As CNN networks are simpler, we were fairly successful in the end as we were able to build on the MobileNetV2 network and add extra layers. In the end this allowed us to achieve an accuracy of 79%. This is quite good considering some classes contained as few as 100 images. Unfortunately, the GAN network was not successful as it required significantly longer to train. As there are many more parameters than the CNN, the exact reason(s) for the lack of success are not clear.

6. REFERENCES

1. A. Quattoni and A. Torralba, "Recognizing indoor scenes," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, 2009, pp. 413-420, doi: 10.1109/CVPR.2009.5206537.

APPENDIX

Best Model Architecture:



Note: Pretrained MobileNetV2 has been hidden for readability purposes. Complete architecture scheme is available on the notebook.