Detecting Turistic Places with Convolutional Neural Networks

Fabricio Torrico-Pacherre, Ian Maguiña-Mendoza and Willy Ugarte[a]
Universidad Peruana de Ciencias Aplicadas (UPC), Lima, Peru

Keywords: VGG16, SOA, Tourism, Application, Place Recognition, Neural Network, Image Processing.

Abstract: A mobile application was developed for the recognition of places from a photo using the technique "content based photo geolocation as spatial database queries". For this purpose, an investigation and analysis of the different existing methods that allow us to recognize images from a photo was carried out in order to select the best possible model and then improve it. Performance comparisons, comparison of number of parameters, Error: imagenet and the Brain-Score were made; once the best model was obtained, the algorithm was implemented and with the results the expected information of the place in the photo was shown. The purpose of this information is to recommend nearby places of interest. In the development stage, first, we implement an architecture with convolutional neural networks VGG16, for the recognition of places, the model was trained, after obtaining a trained model with successful results, the construction phase of the application continued. mobile in order to test the operation of the model. Users will use the app by submitting a photo which will query the trained model, and results will be obtained in seconds, information that will provide a better experience when visiting unknown places.

1 INTRODUCTION

Nowadays, tourists are constantly looking for information about new places and experiences, as well as applications that promote tourism, but these can be somewhat cumbersome to use, or also the new technology implemented in many applications, such as 3D technology, which is very interesting and visual, but also more expensive, so they become unaffordable or have some failure for being a new technology. Therefore, a simple application that contemplates the intuitive and user experience is key to highlight the information that is intended to show. "... New ways of using technology are nowadays within the real reach of cultural tourists, improving interaction and opening new possibilities. In the case of offering a destination, they not only guarantee a return on investment, but are also key tools to promote cultural assets, or to know the visitor's profile, as well as being very useful to achieve excellence in cultural tourism destinations..." (Kontogianni et al., 2022).

In 2019 research was carried out where it was identified that there is a problem with the dissemination of tourist information, since there are no appropriate channels for its presentation or they do not present the clear objectives, in addition, these channels do not provide the current information of the place in a way that is really useful for tourists. They also present a cost for this presentation and this problem usually decreases the interest to visit places for tourist purposes generating a bad experience (Vasconcelos et al., 2021). Computer visual recognition has gained interest in recent years , so advanced deep learning techniques are being employed to address the problem. The achievement of this activity starts from the challenge of addressing various problems such as false recognition of locations that generates interference, which reduces accuracy and causes localization failures, so they must achieve very high recognition accuracy (Masone and Caputo, 2021).

Performing an image recognizer is not usually a simple task, it is a technique that is used by several programs to recognize faces, car license plates, objects, etc. This technique requires hard work to be performed efficiently. The complexity of this task goes hand in hand with what one seeks to detect, since it is necessary to perform a previous training so that the model can automatically recognize an image and, consequently, an extensive data set is required to increase the accuracy of detection, this varies depending on what you want to predict. In the case of detecting a place, it is important to take into account the nature, since it can alter the appearance of the place.

[a] https://orcid.org/0000-0002-7510-618X

Torrico-Pacherre, F., Maguiña-Mendoza, I. and Ugarte, W.
Detecting Turistic Places with Convolutional Neural Networks.
DOI: 10.5220/0010992500003179
In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) - Volume 1*, pages 471-478
ISBN: 978-989-758-569-2; ISSN: 2184-4992

There are models that recognize images for a specific purpose automatically, which use a single image for their operation independent of the data used to train the model, as in the paper Mask R-CNN, in which the authors present a new framework for segmenting object instances. Their approach efficiently detects objects in an image and, at the same time, generates a high-quality segmentation mask for each instance. The method, called Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition (He et al., 2017). These models are not oriented to promote tourism in a direct way and share a common problem which is the accuracy in detecting a place. Therefore, the motivation of this research is to be able to detect a place with a high accuracy, oriented to promote tourism in little recognized places, our research presents the following contributions:

- We perform a comparative analysis of neural network architectures to generate location detection based on metrics of number of parameters, Error: imagenet and Brain-Score.
- We propose a deep learning model, which is based on the use of a VGG16 architecture to perform automatic place detection with a single input image and high accuracy.
- We develop a mobile application which will use the place recognition algorithm to locate on a map where the place is located, as well as surrounding places of interest.

This paper is organized as follows. In section 2 we mention the work related to place detection; in section 3 we will discuss the context on which we based our proposal and explain our contributions in more depth. Subsequently, in section 4 we will present the experiments that were necessary to generate the model. Finally, in section 5 we will specify our main conclusions and results of the finished application.

2 RELATED WORKS/DISCUSSION

The authors of (Zhang et al., 2021) present an introduction to deep learning, specially convolutional neural networks. They then discuss potential deep learning opportunities in visual place recognition (VPR) and present CNN-based place recognition methods using the VGG16 network. With this information, we were able to have a better understanding of how convolutional networks work and how they are applied to image recognition problems, so we were able to adapt them to detect places, starting with the same network presented in VGG16 but applying modifications in its layers to improve its performance.

The authors of (He and Li, 2019) perform a preprocessing of images transformed to gray scale, applying a filter to them. Subsequently they apply a refinement of the image by modifying the brightness, contrast and pixel distribution in order to apply it to their model and obtain a better detection rate. Unlike this previous paper, we did not apply a gray scale to our images because they are in the environment and are affected by it, also because they were limited to images that are illuminated until before sunset, so these images were applied a pre-processing where they were reduced to the same size to all the images in addition to converting them to tensors in order to have a faster and more effective training.

The authors of (He et al., 2020) present a new framework for segmenting object instances. Their approach detects objects in an image and, at the same time, generates a high-quality segmentation mask for each instance to make it easier to detect an image. In contrast, we did not apply object segmentation as it was applied in this paper, because we decided to capture all the information presented in the image and not limit ourselves by specific objects, thus avoiding the generation of a mask for each object that can be seen in the image and accelerating our prediction process.

The authors of (Tian, 2020) present a recurrent neural network in the convolutional neural network and use both to learn the deep features of the image in parallel. In order to establish a dual optimization model, and thus, achieve an integrated optimization of the whole convolution and connection process. In contrast, for the creation of our network we only used the convolutional network because of the previous processing of the images, so that only the use of this network was enough to obtain the necessary features and present satisfactory results (see section 4).

There are various works involving convolutional neural networks for differents tasks (e.g., Natural Language Processing (Trusca and Spanakis, 2020), volumetric representations (Filipovic et al., 2018)), furthermore for enhancing their performance (Al-Hami et al., 2018; Guo et al., 2019), but not that many for place recognition with touristic purposes.

3 PLACE DETECTION WITH DEEP LEARNING

3.1 Preliminary Concepts

In this section we will explain the most relevant topics for this work, about which in section A we will talk about Deep Learning and neural networks, which was the method used to solve the problem. Then in section B we will talk about what is image recognition and what we mean by geolocation. Finally, section C will explain what a software architecture looks like.

3.1.1 Deep Learning - Neural Networks

The field of Deep Learning (DL) is an area integrated to Machine Learning (ML). Within this field is the set of artificial neural networks (ANN), these networks work by learning patterns, which are based on the simulation of the human brain. As its name says, it is a deep learning, this means that it uses computational models that are structured by several layers to perform the learning process, to evaluate and learn the data given to these models (Goodfellow et al., 2016). These data provided for recognition and learning can be numerical, videos, among others. But for the purposes of this work, photos will be used. We can say that DL is supported by ML algorithms, which are based on neural networks (Goodfellow et al., 2016).

These networks show good results when used in learning that requires generating grid representations of data. Convolutional neural networks (CNN) perform well in generating functions and being able to discriminate data correctly. A basic ML system can be seen in Fig. 1. In the first section they generate the data to be used and classify them, in the second section they pre-process the data to be used more efficiently and finally, they divide the data and this is then used in the predictive models. These networks have high performance in the generation and classification of features (Khan et al., 2020).

A typical CNN is comprised of alternating convolution layers and pooling layers followed by one or more fully connected layers. However, sometimes the connected layers are replaced by globally averaged pooling layers. In addition to the basic functions it possesses, CNNs have various regulators such as normalization and dropout functions, which are added to improve network performance (Khan et al., 2020).

VGGs are a type of neural networks, in which their architecture is composed of layers. This type of network was implemented with 19 layers deep in order to perform simulations with a deep search to represent the functioning of a brain neural network.

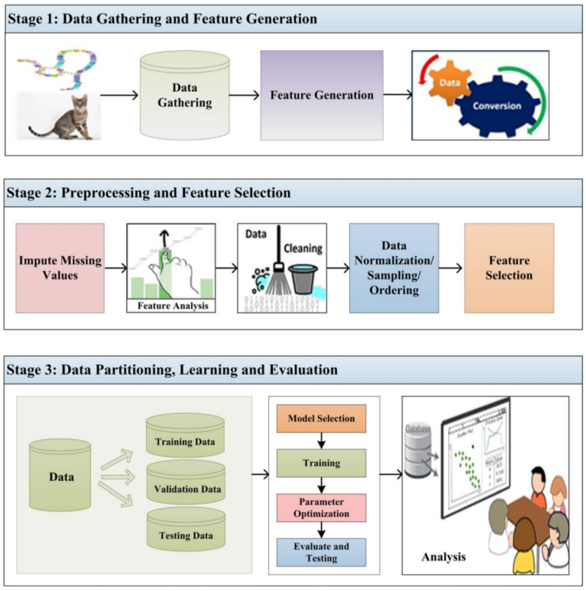

Figure 1: Basic layout of a typical ML system having several stages (Khan et al., 2020).

With the passage of time and new findings, a new filter was implemented, which is used by this network, the VGG replaces the 11x11 and 5x5 filters with a 3x3 filter distributed in its layers, which showed that with a regular filter size (3x3), the effect generated by the 7x7 and 5x5 filters can be replicated. Using a reduced filter size provides a better result with less computational complexity, since the number of parameters is reduced. Thanks to this, there is a tendency to work with smaller filters to use convolutional networks. The VGG uses 1x1 convolutions in its layers, which are convolutional layers, in order to further optimize the complexity of the network. Likewise, these layers learn the different combinations that are generated by the results of the previous layers (Khan et al., 2020). In the Fig. 2 a basic architecture of how a VGG16 neural network is composed can be seen.

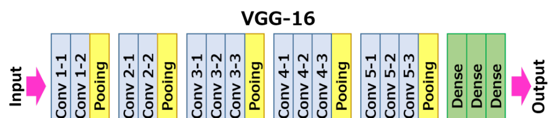

Figure 2: Architecture of a VGG16 network (Pensyl et al., 2019).

3.1.2 Image Recognition - Geolocalization

Image recognition is a technique applied to identify objects or images, where this data is used to generate an accurate decision within a larger system. This technique is used by means of machine learning or deep learning (Heinisch and Ostaszewski, 2018). Image recognition processes the data received through training data and then evaluates it with the test data to get a more accurate response.

According to (Williams et al., 2017), geolocation is understood as the tool capable of allowing the spatial location of any object or person, this tool that uses the coordinates studied by geographers, has been improved over time to the point of creating new applications. This happened with the arrival of mobile devices since it allowed anyone with them to know their location on the map via GPS (Williams et al., 2017).

3.1.3 Software Architecting

Software Architecting or software architecture has had several meanings, one of the last ones is defined as the set of structures necessary to reason about the system comprising parts and relationships between them, in addition to the properties that both have (Knodel and Naab, 2014). It should be considered that when designing a software architecture it should respond directly to the qualitative needs of the client about the system, i.e., it is built from the quality attributes.

A simple example of this is that when requiring mainly that the system is developed in a short time and is not scalable, i.e., more functionalities will not be incorporated later, in addition to being able to test the application easily, it is possible to opt for an architecture that is easy to build such as the layered architecture, which is the best in terms of ease of development and testability. In the Fig. 3 a layered architecture can be seen.

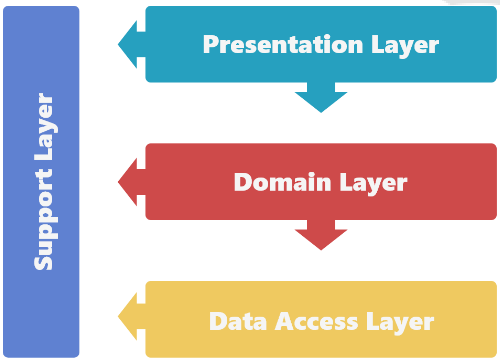

Figure 3: Layered Architecture.

But if, in addition to ease of development, we wish to comply with quality attributes such as scalability or flexibility, we can count on the service-oriented architecture (SOA), as shown in Fig. 4.

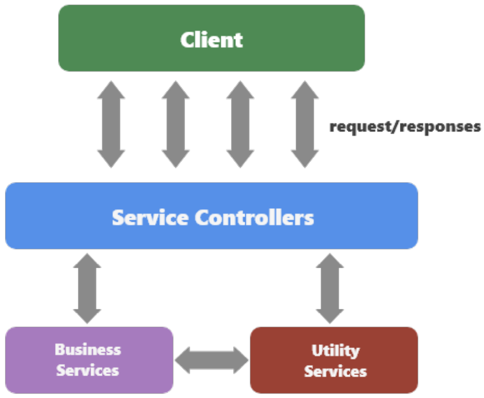

Figure 4: Service Oriented Architecture.

3.2 Method

This section will address the technologies applied in the project for the science and software area, in section A the convolutional model to be applied will be analyzed, and in section B, the topic of the chosen service-oriented architecture.

3.2.1 Convolutional Network

In this section, a comparison of 3 convolutional models was used, as shown in Table 1 bellow, in order to select the best model to be used. This comparison was made in 4 categories:

- Number of parameters: The number of parameters is decided after performing an investigation and making comparisons with the sources found in each network.
- Error: ImageNet: The imageNet error is a test that is usually applied to the CNNs to measure their efficiency, it is through a common dataset and show how much error they have to detect, for this was compared with the sources found in each network and another where the networks are compared.
- N 140: When putting the brain-score, it was decided to put the position where the best type of each network of 140 comparisons is found, in this way to show which is the most outstanding.
- Number of wins: For the calculation of the number of wins, it was decided to use its own criteria where a point would be added for each section where the result is the best, the winner will be the one that has won the most fields

In which the VGG neural network was the winner, the comparison can be seen in the Table 1. This network receives as input a 2D image, where each layer obtains as output the previous layer as input. In each of its convolution layers there is a predetermined filter

Table 1: Models.

	Number of parameters	Error : ImageNet	Brain-Score: N 140	Number of wins
VGG	**138M**	7.3	**2**	**2**
GooLeNet	4M	6.7	36	0
RestNet	25.6M	**3.6**	3	1

and at the time of performing the convolution operation, each filter is displaced in the input by the number of jumps, since the data obtained by each image usually have a higher correlation in a local area as opposed to a global area. In this convolution process all the features of the input image to which the filters are applied are extracted, so this extracted data is called feature map (Ha et al., 2018).

After extracting the features from the image, the clustering layer gathers the similarities found for each feature and thanks to this the performance of the network becomes distortion invariant, furthermore in this layer it also has the function of reducing the dimensions of the feature map and when this resulting map enters the next layer a new one is generated (Ha et al., 2018). After starting with the base configuration of the network, modifications were made with the fine-tuning function of the torchvision library, by means of which the values of the last 4 layers were altered to try to obtain better results than the base network in the prediction of a location.

3.2.2 Service Oriented Architecture

The service-oriented architecture or SOA corresponds to a style of software architecture that consists of thinking in services for each aspect of the system, which will be designed to be reusable through public interfaces, allowing the existence of an ecosystem of interaction between providers and consumers. SOA is implemented based on 4 abstractions:

- Services are the business logic and data management, contemplating access restrictions and an interface with which to expose them to consumers.
- The frontend is the application or UI, which interacts directly with the user.
- The service repository is where the designed services will be hosted, i.e. the cloud, to be available to the allowed users accessing the Internet
- The service bus is the way in which the services communicate with the consumers (Avila et al., 2017).

The reason to take into account this architecture is for the very fact of having independent services, from this we can create a system and modify it continuously because programming a new functionality does not cause conflict problems with the previous ones, then meet quality attributes such as scalability or testability. On the frontend side, the only concern is to consume the service correctly, pointing to the correct endpoint, in our case from the mobile application. The following is a container diagram (see Fig. 5), part of the C4 model[1]used to explain how the chosen architecture will be adapted to our project, which shows the interaction between the user and the application, which, in turn, interacts with the APIS "Management" and image recognition "API Recognition System", both containers interact with the MySQL database. All these containers are managed by the Cloud services environment, Amazon Web Services.

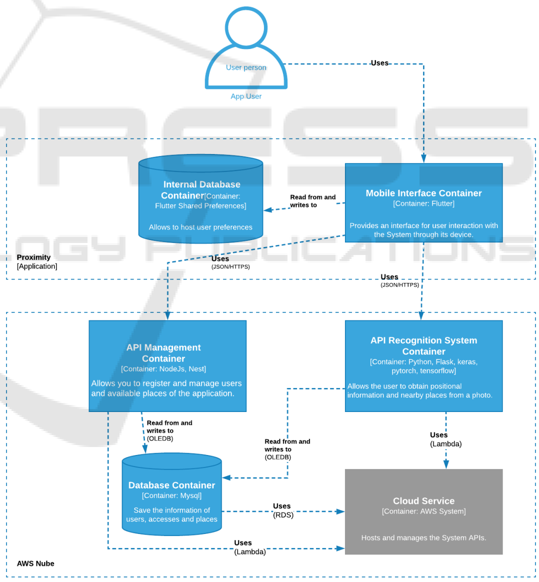

Figure 5: C4: Project Container Diagram.

4 EXPERIMENTS

In this section we describe the working environment of the experiments performed, the process carried out for the collection of information needed to train the model, the image pre-processing and the model created with the various features it contains. Then we developed our application.

[1]C4 model - https://c4model.com/

4.1 Experimental Protocol

- **Working Environment:** To carry out the experiments a series of decisions were made and frameworks that were indispensable were adopted.
  1. We made use of the free version of the google collab platform, with which we developed in Python version 3.8 and mainly used the pytorch libraries, which helped us to generate a better training of the convolutional model. On the services side, which were necessary to communicate the interfaces with the database and/or the model, Flask in version 2.0.1 was adopted to access the recognition algorithm developed in collab. In VS Code the libraries that were installed in the local environment were: flask, pytorch, pandas, folium and numpy, to be able to execute it as API.
  2. For the management services, we used NodeJs v14.15.1 and Nest 4.6.0, in the windows 10 operating system, which is not determinant because the mentioned frameworks are multi platform. In addition, git was in- stalled in order to manage repositories and have a version control.

  Finally, for the construction of the application we used Flutter v2.2.3, Dart 2.13.3
- **Compilation of Images:** In order to feed the model, it was necessary to obtain 2400 images of the 4 places that would be part of our test, located in the district of San Miguel, Lima - Peru. The images would have to be processed in order to improve the study.

The source code of the project can be found at https://gitlab.com/researchgames_upc

4.2 Results

As previously mentioned, tests were performed on the last 4 layers of the neural network in order to obtain better results, 5 runs were performed on the modified layers where a slight improvement in the prediction of the network was evidenced, these results can be seen in Table 2. For the following comparative tables, the best results obtained according to the compared category will be taken, however, in the case of acurracy, the highest will be taken and the winning results will be painted in bold.

- Model 1: It represents the base configuration of the neural network.
- Model 2:
  - Layer 30 containing kernel size: 2, stride: 1, padding: 1, dilation: 4 and ceil mode: FALSE was altered.
  - Layer 29 containing inplace: FALSE.
  - Layer 28 containing kernel size: 1,1 ; stride: 2,2 and padding: 3,3.
  - Finally layer 27 containing inplace: FALSE.
- Model 3:
  - Layer 30 containing kernel size: 2, stride: 3, padding: 0, dilation: 3 and ceil mode: TRUE.
  - Layer 29 containing inplace: TRUE.
  - Layer 28 containing kernel size: 2,2 ; stride: 3,3 and padding: 2,2.
  - Finally layer 27 containing inplace: TRUE.
- Model 4:
  - Layer 30 containing kernel size: 2, stride: 2, padding: 0, dilation: 1 and ceil mode: FALSE was altered.
  - Layer 29 containing inplace: FALSE.
  - Layer 28 containing kernel size: 2,2 ; stride: 3,3 and padding: 0,0.
  - Finally layer 27 containing inplace: FALSE.
- Model 5:
  - Layer 30 containing kernel size: 2, stride: 1, padding: 1, dilation: 2 and ceil mode: TRUE.
  - Layer 29 containing inplace: TRUE.
  - Layer 28 containing kernel size: 1,1 ; stride: 2,2 and padding: 3,3.
  - Finally layer 27 containing inplace: TRUE.

Table 2: Fine Tunning.

	Time	Accuracy
Model 1	1h 20m 20s	0.979
Model 2	1h 20m 50s	0.988
Model 3	**1h 19m 25s**	0.985
Model 4	1h 22m 28s	0.979
Model 5	1h 23m 20s	**0.989**

It was necessary to compare these 5 models in order to find out which configuration showed the best results and thus choose the best model for the research. To verify the results obtained with our model we proceeded to perform new tests with the same dataset obtained, these tests were performed on 3 different algorithms to demonstrate the efficiency of neural networks, these algorithms were decision trees, support vector machines (SVM) and GradientBoosting. For each of these algorithms, an investigation was first carried out to see how they work and to be able to find a base structure in which it is easy to modify and attach the image dataset.

Table 3: Classification methods.

(a) Decision Trees.

	Fixed size	Bins	N trees	Seed	Time	Accuracy
DT 1	500, 500	8	100	9	18m55s	**.999**
DT 2	500, 500	16	100	9	19m45s	**.999**
DT 3	400, 400	8	100	9	25m37s	**.999**
DT 4	400, 400	16	100	9	**9m53s**	**.999**
DT 5	400, 400	32	100	9	10m24s	.649

(b) Support vector machines.

	C	Gamma	Kernel	Time	Accuracy
SVM 1	0.1	.0001	rbf	1h29m35s	.970
SVM 2	0.1	.0001	poly	**34m19s**	.995
SVM 3	0.1	.0010	rbf	2h13m52s	.885
SVM 4	1.0	.0001	rbf	36m36s	**.100**
SVM 5	1.0	.0001	poly	24m32s	**.100**

Table 4: Gradient Boosting.

	Learning rate	Time	Accuracy
GradientBoosting 1	.30	1.10s	.972
GradientBoosting 2	.35	1.10s	.970
GradientBoosting 3	.40	1.10s	.973
GradientBoosting 4	.45	0.99s	.978
GradientBoosting 5	.50	0.99s	**.980**
GradientBoosting 6	.55	**0.98s**	.980
GradientBoosting 7	.60	**0.98s**	.982
GradientBoosting 8	.65	0.99s	.975
GradientBoosting 9	.70	0.99s	.977
GradientBoosting 10	.75	**0.98s**	**.980**

In the Table 3 the values of the fixed size variable were modified, which modified the size of the images in order to have a standard at the moment of being trained and the bins, which are a small set of data, in this case of images.

In the Table 3b the values C (which controls the cost of the calculation errors), gamma (which varies the distribution of the data assigned to a support vector) and kernel (which is a variable where the mathematical function is defined) were modified.

In the Table 4 the learning rate is altered, it is the hyperparameter/variable that controls how fast a Gradient Boosting algorithm learns, it is known that it can limit the possibilities of overfitting.

4.3 Discussion

Table 2 shows the results of altering the neural network to obtain better results, as can be seen the test 05 showed the best percentage when detecting an image so we decided to establish it as the final configuration of the network. As a result of these experiments, we obtained that the algorithm of decision trees which are seen in Table 3, presented certain inferior results compared to the neural networks at the time of detecting an image, however it was evidenced in the last tests that the percentage when detecting a place exceeds 98. Despite this, a convolutional network is still the best option for these problems since, unlike the decision trees, is perfectible with more data, so if classes are increased, in our case places, it will have a better accuracy since it works deeper than this algorithm.

Similarly, the SVM algorithm shown in the Table 3b, presents a similar behavior, since not in all the results obtained it exceeds that of our network, however, unlike the decision trees, this algorithm works faster with few classes so more tests could be performed in less time. Even so, neural networks remain as the best option due to the same problem, performance, as it does not improve by increasing the number of classes.

Finally, the GradientBoosting algorithm, which can be seen in the Table 4, was the fastest in execution, the speed shown when detecting the consulted place was considerably faster than the other algorithms so that a greater number of comparative tests could be performed, however, these results do not exceed the proposed network, so with the answers obtained it is evident that this algorithm is inferior to ours. These executions allow us to confirm that, for this type of problems, using the convolutional neural network is one of the best options w.r.t. performance.

5 CONCLUSIONS AND PERSPECTIVES

We conclude that the algorithm belonging to the VGG16 model is applicable in the test sites within Lima, parks and ruins, in the future should be considered more spaces with topographic variations and conduct more experiments throughout the country, in order to ensure the result. Evenmore, this kind of architecture was applied to other kinds of problems, for instance sheet music recognition (Lozano-Mejía et al., 2020) or fruit ripeness (Rodríguez et al., 2021).

We consider that the software architecture used, SOA or Service Oriented Architecture, was the most appropriate because it provides a scalable environment to the project, that is, it allows the project to remain current with the continuous incorporation of users, as opposed to other monolithic architectures that do not, or others that being scalable are more complex as micro services, which is more suitable for much larger projects.

In next steps we would like to incorporate the possibility of feeding the model, but this time by other users that through the continuous input of images open new places of reference that the application can detect allowing the application to evolve over time.

REFERENCES

Al-Hami, M., Pietron, M., Casas, R. A., Hijazi, S. L., and Kaul, P. (2018). Towards a stable quantized convolutional neural networks: An embedded perspective. In *ICAART*.

Avila, K., Sanmartin, P., Jabba, D., and Jimeno, M. (2017). Applications based on service-oriented architecture (SOA) in the field of home healthcare. *Sensors*, 17(8).

Filipovic, M., Durovic, P., and Cupec, R. (2018). Experimental evaluation of point cloud classification using the pointnet neural network. In *IJCCI*.

Goodfellow, I. J., Bengio, Y., and Courville, A. C. (2016). *Deep Learning*. Adaptive computation and machine learning. MIT Press.

Guo, Y., Cao, H., Bai, J., and Bai, Y. (2019). High efficient deep feature extraction and classification of spectral-spatial hyperspectral image using cross domain convolutional neural networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, 12(1):345–356.

Ha, I., Kim, H., Park, S., and Kim, H. (2018). Image retrieval using bim and features from pretrained vgg network for indoor localization. *Building and Environment*, 140.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. (2017). Mask R-CNN. In *IEEE ICCV*.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. (2020). Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(2).

He, T. and Li, X. (2019). Image quality recognition technology based on deep learning. *J. Vis. Commun. Image Represent.*, 65.

Heinisch, P. and Ostaszewski, K. (2018). Matcl: A new easy-to use opencl toolbox for mathworks matlab. In *ACM IWOCL*.

Khan, A., Sohail, A., Zahoora, U., and Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.*, 53(8).

Knodel, J. and Naab, M. (2014). Software architecture evaluation in practice: Retrospective on more than 50 architecture evaluations in industry. In *IEEE WICSA*.

Kontogianni, A., Alepis, E., and Patsakis, C. (2022). Promoting smart tourism personalised services via a combination of deep learning techniques. *Expert Syst. Appl.*, 187.

Lozano-Mejía, D. J., Vega-Uribe, E. P., and Ugarte, W. (2020). Content-based image classification for sheet music books recognition. In *IEEE EirCon*.

Masone, C. and Caputo, B. (2021). A survey on deep visual place recognition. *IEEE Access*, 9.

Pensyl, W. R., Min, X., and Lily, S. S. (2019). Facial recognition and emotion detection in environmental installation and social media applications. In *Encyclopedia of Computer Graphics and Games*. Springer.

Rodríguez, M., Pastor, F., and Ugarte, W. (2021). Classification of fruit ripeness grades using a convolutional neural network and data augmentation. In *FRUCT*.

Tian, Y. (2020). Artificial intelligence image recognition method based on convolutional neural network algorithm. *IEEE Access*, 8.

Trusca, M. M. and Spanakis, G. (2020). Hybrid tiled convolutional neural networks (HTCNN) text sentiment classification. In *ICAART*.

Vasconcelos, C., Reis, J. L., and Teixeira, S. F. (2021). Use of mobile applications in the tourism sector in portugal - intention to visit the algarve region. In *WorldCIST*.

Williams, E., Gray, J., and Dixon, B. (2017). Improving geolocation of social media posts. *Pervasive Mob. Comput.*, 36.

Zhang, X., Wang, L., and Su, Y. (2021). Visual place recognition: A survey from deep learning perspective. *Pattern Recognit.*, 113.