



A Deep learning integrated mobile application for historic landmark recognition: A case study of Istanbul

Bulent Bayram ¹, Batuhan Kilic*¹, Furkan Ozoglu ², Firat Erdem ³, Tolga Bakirman ⁴, Sinan Sivri ⁴, Onur Can Bayrak ¹, Ahmet Delen ⁵

¹ Yildiz Technical University, Civil Engineering Faculty, Department of Geomatic Engineering, Istanbul, Turkey

² Istanbul Metropolitan Municipality, Directorate of Geographic Information Systems, Istanbul, Turkey

³ Eskisehir Technical University, Earth and Space Sciences Institute, Eskisehir, Turkey

⁴ Istanbul Technical University, Research and Application Center for Satellite Communications and Remote Sensing, Istanbul, Turkey

⁵ Gaziosmanpasa University, Department of Geomatic Engineering, Tokat, Turkey

Keywords

Deep learning
Convolutional Neural
Network (CNN)
Historic landmark
recognition
Mobile technology

ABSTRACT

Recent developments in mobile device technology and artificial intelligent systems took the attention of many researchers. Historical sites and landmarks are the indispensable heritage of cities. Historic landmark recognition, including detailed attribute information, can connect people directly with the history of the cities, although they may not be familiar with the impressive historical monument. This can be achieved by integrating mobile and deep learning technologies. Therefore, we focused on establishing a deep learning (DL) based mobile historic landmark recognition system in this study. The VGG (16, 19), ResNet (50, 101, 152), DenseNet (121, 169, 201) DL architectures were trained by end-to-end learning techniques for the recognition of ten historic landmarks from the metropolitan city of Istanbul, Turkey. The dataset was prepared by collecting images of ten historical buildings from the image hosting services. The developed prototype automatically and instantly recognizes these historic landmarks from scene images and immediately provides related historic information as well as route planning. The experimental results indicate that DenseNet-169 architecture is very effective for our dataset with 96.3% accuracy. This study has shown that deep learning offers a promising alternative means of recognizing historic landmarks.

1. INTRODUCTION

Istanbul is one of the most significant metropolitan cities in the world. This city, which hosts the unique signs of European and Asian communities through Byzantine and Ottoman cultural heritage, has been named as the 2010 European Capital of Culture due to its unique historic areas that attract many visitors (UNESCO, 2006).

Tourism is the name given to trips to an unknown place (Brown, 2007; Mulazimoglu and Basaraner, 2019). Visitors aspire to discover and witness the varied life of the city (Richards, 2018). In this respect, landmarks, known as recognizable natural or artificial features, often

attract people to visit and create memories to share with their social group. Sometimes during a trip, unknown historical objects can attract attention (Cheng and Shen, 2016). However, they can only interact visually with cultural objects without detailed information. This situation results in a limited awareness of cultural heritage. To improve visitors' experience, rapid, accurate, significant, and real-time information is needed.

The growing use of social networks has provided large amounts of data relating to every field. This has brought new opportunities for concept-based image recognition (Simonyan and Zisserman, 2014). People

Cite this article

* Corresponding Author

(bayram@yildiz.edu.tr) ORCID ID 0000-0002-4248-116X
*(batuhank@yildiz.edu.tr) ORCID ID 0000-0002-0529-8569
(furkan.ozoglu@ibb.gov.tr) ORCID ID 0000-0002-6276-3762
(firaterdem@eskisehir.edu.tr) ORCID ID 0000-0002-6163-1979
(tolga@cscrs.edu.tr) ORCID ID 0000-0001-7828-9666
(sinan@cscrs.edu.tr) ORCID ID 0000-0002-8591-9555
(onurcb@yildiz.edu.tr) ORCID ID 0000-0002-5147-747X
(ahmet.delen@gop.edu.tr) ORCID ID 0000-0003-3091-8501

Bayram B, Kilic B, Ozoglu F, Erdem F, Bakirman T, Sivri S, Bayrak O C & Delen A (2020). A Deep learning integrated mobile application for historic landmark recognition: A case study of Istanbul. Mersin Photogrammetry Journal, 2(2), 38-50

have started voluntarily to share their images using different social media applications such as Flickr, Facebook, and Instagram (Weyand and Leibe, 2015) and internet search engines. A variety of studies and applications have been carried out, investigating location estimation (Hays and Efros, 2008), scene recognition (Zhou et al., 2018), face recognition (Parkhi et al., 2015) and landmark recognition (Cheng and Shen, 2016), using images taken from these kinds of a large database.

The enhancement and promotion of cultural heritage using information and communication technologies are an essential research issue (Amato et al., 2017; Şasi and Yakar, 2018). Although human-computer interaction and mobile digital technology have the potential to provide to access cultural heritage information (Doğan and Yakar, 2018; McGookin et al., 2019), existing frameworks may not be capable of presenting smart and detailed data. The state-of-art DL techniques brought new opportunities to overcome this problem. DL is based on multi-layer Artificial Neural Network (ANN) and a subset of machine learning. It has been confirmed in many studies such as computer vision, image classification, robotics, bioinformatics, biomedical, geomatics (Zhang et al., 2018) that DL has become a handy tool for image and information analysis, primarily using open databases and internet resources (Tzelepi and Tefas, 2018).

Jiang et al. (2017) developed a real-time internet cross-media retrieval system using DL. Shukla et al. (2017) proposed a deep convolutional neural networks model for the recognition of 117 Indian monuments. Termritthikun et al. (2018) proposed a DL network named as NU-LiteNet for mobile landmark recognition.

Huang et al. (2018) proposed DL as a tool to solve multi-concept-based image retrieval problems using MIR Flickr 2011 and NUS-WID dataset. F. Huang et al. (2018) combined content and network for multi-view learning and proposed a deep multi-view embedding model-based image recognition system. Xu et al. (2019) used the same dataset and developed a unimodal semantic image retrieval system using DL techniques.

As can be seen from the literature review, there is limited study concerning DL for mobile historic landmark recognition. Therefore, the objective of this study is to test the efficiency of different DL architectures for mobile historic landmark recognition. For this purpose, the Visual Geometry Group (VGG16, 19) (URL1), Deep Residual Network (ResNet-50, 101, 152) (URL2), and Densely Connected Convolutional Network (DenseNet-121, 169, 201) (URL3) architectures were exploited to recognize ten selected historic landmarks in Istanbul, Turkey. All utilized DL architectures have been trained and tested with our generated dataset.

2. STUDY AREA AND DATASET

The generated dataset consists of the most popular and essential ten historic landmarks within the boundaries of Istanbul. These are the Maiden's Tower, the Sultan Ahmet Mosque (Blue Mosque), the Galata Tower, Hagia Sophia, the Ortaköy Mosque, the Topkapi Palace, the Valens Aqueduct, the Dolmabahce Palace, the Obelisk of Theodosius, and the Dolmabahce Clock Tower (Table 1). The distribution of landmarks can be seen in Figure 1.



Figure 1. Distribution of landmarks

Table 1. Samples of the selected historical landmarks

Landmarks	Construction	Name	Description
1 	Stone (concrete) tower, masonry cupola	Maiden's Tower	It is built up on the rock, 200 meters from the coast of Üsküdar, Istanbul, on the Bosphorus. It was built around 419 BC (IPDCT, 2019a).
2 	Mosque (Islamic, Late Classical Ottoman)	Blue Mosque	This unique mosque in Istanbul has six minarets and eight domes. It was built by Sedefkar Mehmet Aga between 1609–1616 (IPDCT, 2019b).
3 	Stone tower	Galata Tower	It was built in the 14th century by the Genoese (IPDCT, 2019c).
4 	Mosque (museum)	Hagia Sophia	It was built between 537 and 537 AD and was famous in particular for its massive dome (IPDCT, 2019d).
5 	Mosque (Baroque Revival)	Ortaköy Mosque	It was built in 1853 by architect Karabet Balyan and was then repaired between 1960 and 1972 (IPDCT, 2019e).
6 	Palace (Various low buildings, surrounding courtyards, pavilions and gardens)	Topkapi Palace	It was built between 1466 and 1478 by the sultan Mehmet II. The palace was the political centre of the Ottoman Empire between the 15th and 19th centuries (IPDCT, 2019f).
7 	Arch bridge (stone, brick)	Valens Aqueduct	It is part of Istanbul's ancient water supply system and was built in the late 4th century AD during the reign of Roman Emperor Valens (Yorulmaz and Çelik, 2015).
8 	Palace (Baroque, Rococo, Neoclassical)	Dolmabahçe Palace	It was built by order of Sultan Abdulmecid I between 1843 and 1856 (IPDCT, 2019g).
9 	Column (granite)	Obelisk of Theodosius	The obelisk was built by Pharaoh Thutmose III (1479–1425 BC) during the 18th dynasty (IPDCT, 2019h).
10 	Clock tower (Ottoman Neo-Baroque)	Dolmabahçe Clock Tower	Built in 1890–1895, by palace architect Sarkis Balyan II. It was built for Sultan Abdulhamid between 1842–1918 (IPDCT, 2019i).

Two different datasets were prepared to train all the selected DL architectures from the Bing, Foursquare, and Yandex web platforms. The first one (Istanbul-2500) consisted of 2500 images and the second one (Istanbul-5000) 5000 images of all selected historic landmarks. In addition to the online image resources, 310 new images were taken in the field for the Istanbul-5000 dataset. The images which have the same width and height were selected to avoid possible image deformations and resized to 224 x 224 pixels as input dimensions of used DL architectures.

3. METHODOLOGY

Due to their proven success in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Deng et al., 2012), the VGG (16 and 19) (Simonyan and Zisserman, 2014), ResNet (50, 101 and 152) (He et al., 2016a), and DenseNet (121, 169 and 201) (Huang et al., 2017) DL architectures were utilized for this study. These DL architectures have also been used in many other image

classification studies (Nawaz et al., 2018; Nibali et al., 2017; Rothe et al., 2018). All the used architectures were tested for recognition of selected historic landmarks. Our mobile application was integrated with the DL architecture that gave the best accuracy. A flowchart of the study is given in Figure 2.

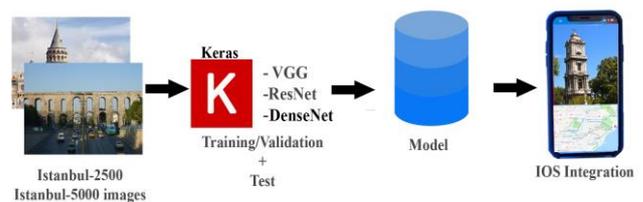


Figure 2. The flowchart of the study

3.1. Implementation of Different Deep Learning Architectures

The details of all the deep CNN architectures employed for historical object recognition using the

Istanbul-2500 and Istanbul-5000 datasets are given below:

3.1.1. VGGNet

The VGG neural networks were developed by the Visual Geometry Group. There are four different versions of the VGG neural networks, which include different weight layers of 11, 13, 16, and 19, respectively (Simonyan and Zisserman, 2014). In this study, the VGG16 and VGG19 versions were employed. The VGG neural networks consist of convolution, pooling, and fully-connected layer and softmax layers. The filter kernel size in the convolution layers is 3 x 3. The rectified linear unit (ReLU) activation function is applied for the nonlinearity process after each convolution layer. After each convolution layer, there are a max-pooling layer with 2 x 2 window sizes (Patterson and Gibson, 2017). In the pooling layers, the shifting interval of the filter is defined by a stride parameter. The stride parameter for VGG neural networks is defined as two. Three fully-connected layers and Softmax classifier layers are used in the last part of the architecture. The first two fully-connected layers have 4096 channels. Dropout regularization is applied in these layers to avoid overfitting problems. A dropout ratio of 0.5 is selected. The amount of the channels in the last fully-connected layer is equal to the count of the classes (Simonyan and Zisserman, 2014). The structures of the VGG architectures used in this study are given in Figure 3.

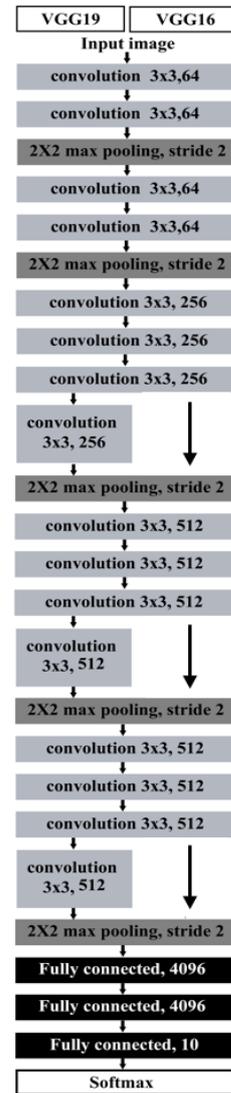


Figure 3. VGG16 and VGG19 layers (Simonyan and Zisserman, 2014)

3.1.2. ResNet

ResNet was developed by the Microsoft research team to reduce the training difficulty of deeper neural networks. The main idea of ResNets is to learn the additive residual function using an identity mapping by using short connections (He et al., 2016b). It has versions consisting of 18, 34, 50, 101, and 152 weight layers (He et al., 2016a). Instead of learning non-discriminatory functions in ResNet architectures, residual functions are adopted using input layers. Unlike VGG, ResNet architectures have shortcut connections which are used in feed-forward neural networks. Thus, shortcut links do

not contain extra parameters and do not cause computational complexity. In this way, relevant information from the previous layer can be transferred to the next layers (He et al., 2016b).

In contrast to VGG architectures, ResNet architectures contain a global average pooling layer and a fully connected layer at the end of the network. Without a dropout operation, the average value in each property map is transferred to the next layer in the global average pooling process (Lin et al., 2013). The parameters of the ResNet layers used in this study are given in Figure 4.

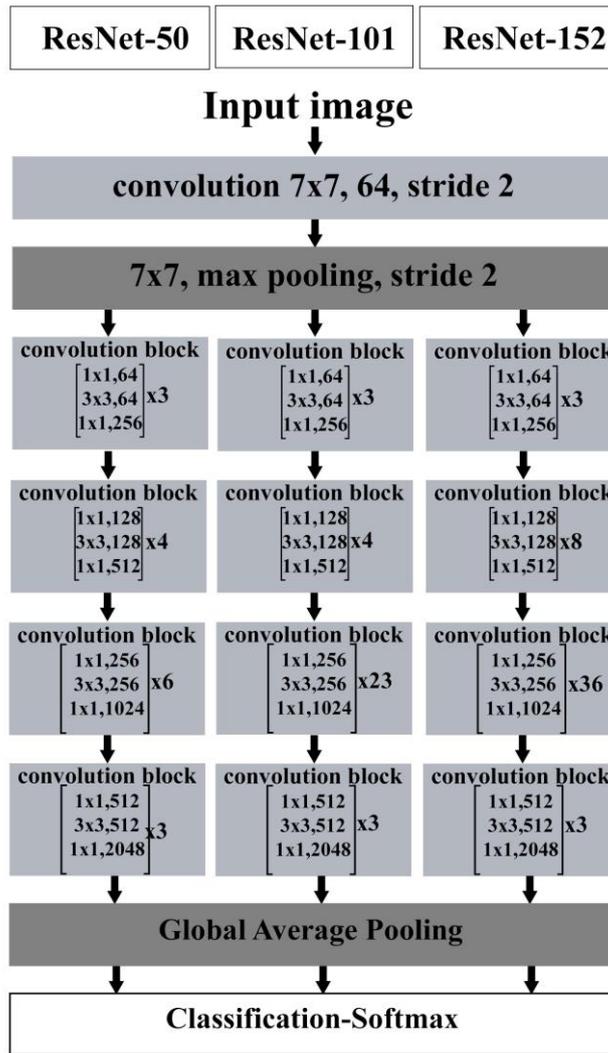


Figure 4. ResNet50, ResNet101 and ResNet152 layers (He et al., 2016a)

3.1.3. DenseNet

DenseNet is well known CNN architecture. Individual layers are connected with every layer behind it. Thus, all layers are used for a decision instead of the single final layer by CNN. DenseNet has been developed by the inspiration of ResNet18 architecture, which is one of the best deep learning architecture and used in many different image classification related studies. Compared to ResNet architecture, it has been seen that if shorter connections between layers close to the input and layers close to the output, the training accuracy can be obtained more accurate. DenseNet architecture is based on the feed-forwarded connection of each layer to every other layer to extend the shorter connections. By DenseNet, feature propagation is reinforced, the amount of parameters is diminished (Gunawan et al., 2018).

DenseNet also has different versions, consisting of 121, 169, 201, and 264 weight layers. Similarly to ResNets, DenseNets use a block concept, too. Unlike ResNets, the principle of intense connections is used in the blocks, according to which each layer in a block has a connection with the previous layers. These connections are provided by transferring feature maps. There are convolution and pooling layers between the blocks. In DenseNet architectures, except for the first convolution layer, dropout is applied with a 0.2 dropout rate after each convolution layer. In addition to this, the ReLU activation function is used following the convolution layers (Huang et al., 2017). The parameters and structure of DenseNet architectures can be seen in Figure 5.

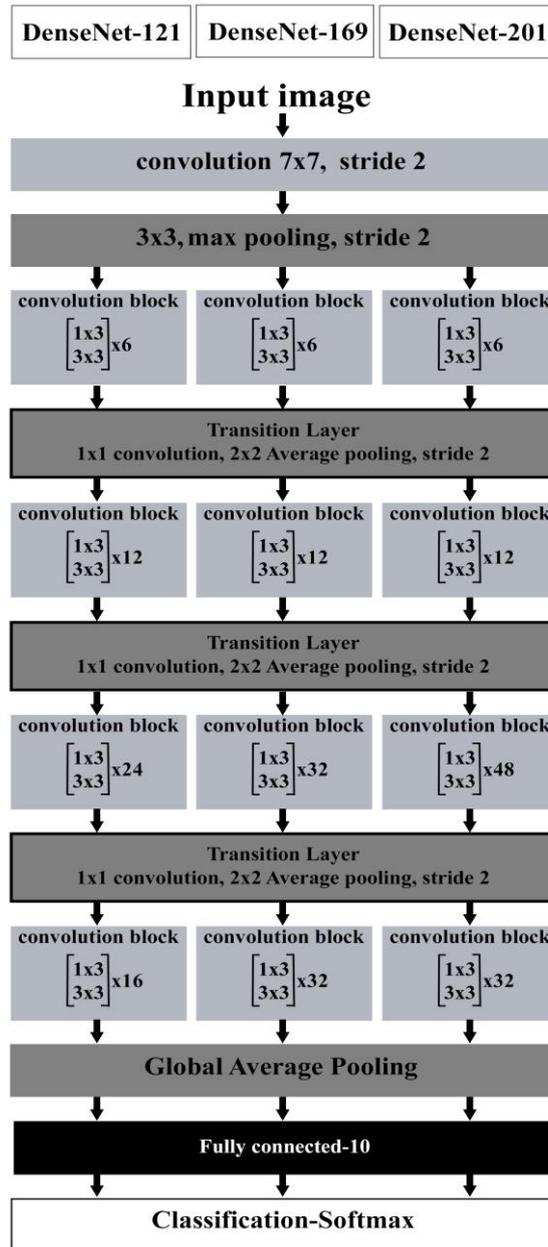


Figure 5. DenseNet121, DenseNet169 and DenseNet201 layers (Huang et al., 2017)

Table 2. Used hyperparameters

Deep Learning Models	Base Learning Rate	Rho	Epsilon	Decay	Epoch	Mini-Batch Size
VGG16	0.05	0.95	1e-07	0	120	16
VGG19						
ResNet50	1	0.95	1e-07	0	120	16
ResNet101						
ResNet152						
DenseNet121						
DenseNet169						
DenseNet201						

Used hyperparameters are given in Table 2. The base (initial) learning rate affects the optimization parameters in order to minimize CNN estimation errors. ρ is a constant, which controls the decay of parameter updates (Patterson and Gibson, 2017). Epsilon is a floating-point number and very close to zero. It is used to prohibit mistakes similar to dividing by zero. Decay is the

rate of initial learning (Keras, 2019). Epoch is the number of training iterations (Patterson and Gibson, 2017). The mini-batch size is the number of records that are passed into a defined learning algorithm at the same time (Soon et al., 2018). Components of used hardware and software specifications can be seen in Table 3.

Table 3. Specifications of hardware and software used

Hardware			
Computer	Desktop PC	MacBook Pro Retina (Mid 2012)	
CPU	Intel® Core™ i7-8700K 3.7GHz	Intel® Core™ i7-3615QM 2.3GHz	
Display Card	NVIDIA GeForce GTX1080Ti 11 GB	NVIDIA GeForce GT 650M 1 GB	
Memory	32 GB	8 GB	
Software			
Library	Tensorflow	Keras	CoreML
Version	1.5.0	2.1.3	2.0

3.2. Integration of the Mobile Application

The iOS platform, one of the most popular mobile application platforms in the world, is continuously being updated for iPhone and iPad devices. The CoreML framework structure developed by Apple for machine-learning applications was announced to developers in 2017 (CoreML Framework, 2019). Also, the CoreML tools that work with Python were used to transfer other machine and DL frameworks (i.e., Keras, Tensorflow, IBMWatson) to mobile applications (CoreML Documentation, 2019). The Vision Framework is integrated with the CoreML library and is used to define images via a mobile application (Maskrey & Wang, 2018). The free Mapkit Framework library was developed by Apple to create map-based mobile applications (MapKit Framework, 2019). It has two modes, namely standard, and satellite. The Core Location Framework (Core

Location Framework, 2019) works by being integrated with the GNSS receiver on a mobile device. The device’s location can be seen on the map when it is used with MapKit. The mobile phone integration flowchart is given in Figure 6. The application for mobile phone integration was developed by Xcode 10 IDE (Integrated Development Environment) using the Swift 4.2 environment. The developed application only works with iPhone instruments and supports iOS Versions 10.0–13.0. In addition to the integration of the Vision, Core Location and MapKit frameworks to allow the use of trained deep CNN models offline, the mlmodel was created for mobile phones by exploiting the CoreML framework. However, for route planning and location definition, the navigation services in the application need access to a network (3G/4G) or WiFi connection.

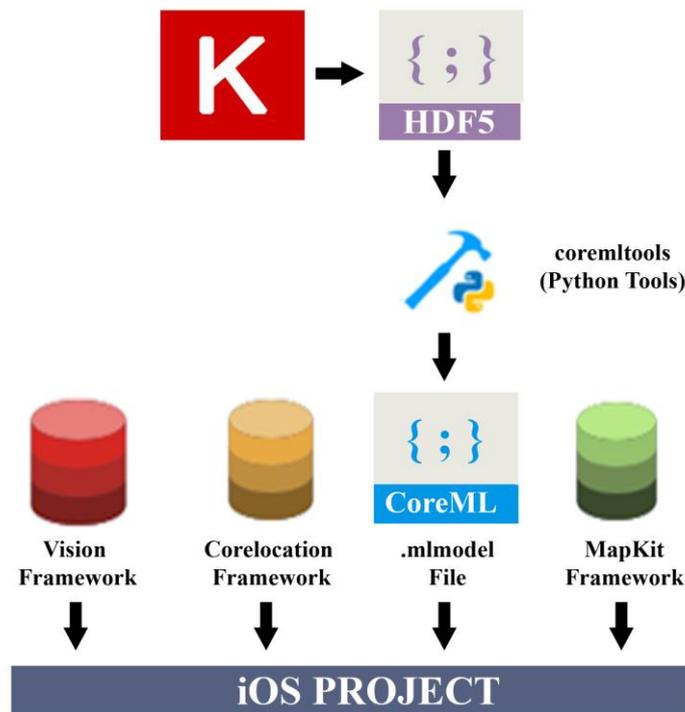


Figure 6. Mobile integration flowchart

4. RESULT AND DISCUSSION

In this study, we investigated the performance of 8 deep CNN architectures (VGG16, VGG19, ResNet50, ResNet101, ResNet152, DenseNet121, DenseNet169

and DenseNet201) for mobile historic landmark recognition. Python’s Keras library (Chollet, 2015) was used to implement all of the selected deep CNN

architectures. 70% of the dataset was used for training, 10% for validation, and 20% for testing. The highest test accuracy was achieved by the DenseNet169 architecture for the Istanbul-5000 dataset (96.3%).

The robustness of the optimization algorithms is still controversial, and there is no consensus regarding the optimal optimization algorithms among experts (Schaul et al., 2014). Therefore, training of the deep CNNs was carried out using the Adadelta optimization algorithm. This method uses only first-order information during dynamical adaption and minimal computational cost. It is robust to noisy gradient

information and accommodates a selection of hyperparameters (Zeiler, 2012).

The training/validation accuracy for each epoch was compared to evaluate the training. Therefore, the training accuracy of each deep CNN architecture and dataset were analysed separately. As can be seen in Figure 7, the test accuracy for VGG16 and VGG19 with two datasets reached a maximum of around 80%. This shows that the VGG16 and VGG19 architectures are unable to produce a comprehensive solution for this study.

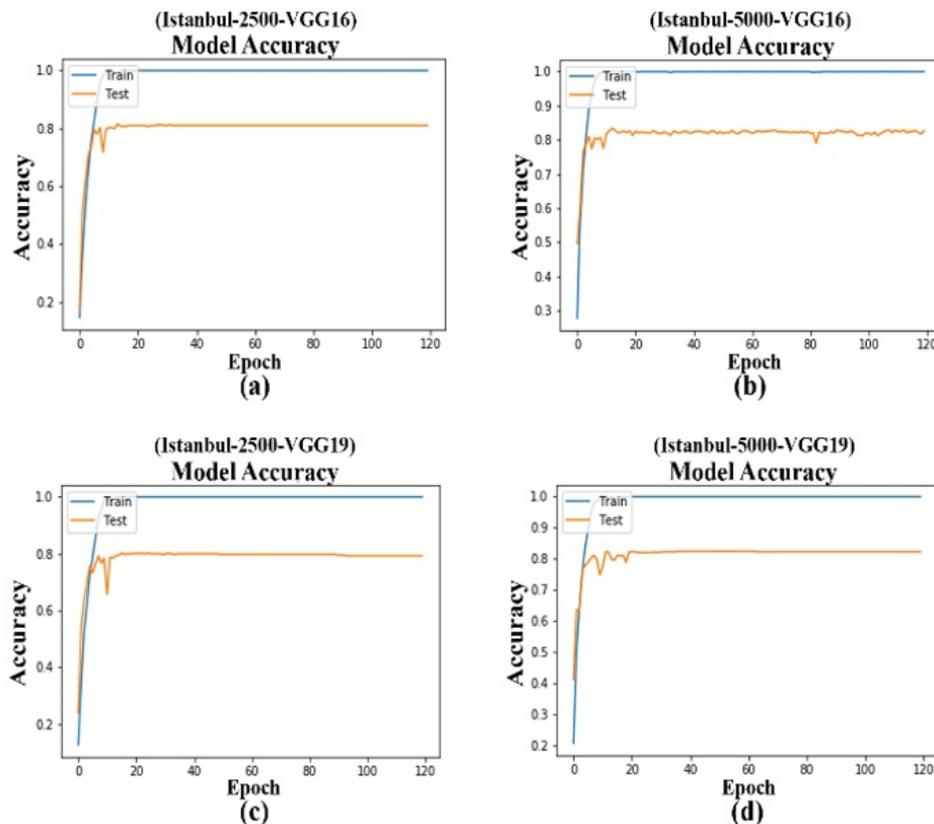


Figure 7. VGG16 and VGG19 training results for the Istanbul-2500 and Istanbul-5000 datasets

The training results for the ResNet architectures (50, 101, and 152) were much noisier than those for the DenseNet models (Figure 8 and Figure 9). It has been observed that, in the epochs where the accuracy of training increased, the test accuracy showed opposite directions for all ResNet architectures on both datasets. However, in general, in the epochs where training was completed, training and test accuracies converged (Figure 8). As a result, the calculated accuracies for the ResNet architectures were similar to those for the VGG architectures.

The evaluation of the training/test accuracies for the DenseNet architectures (121, 169, and 201) for the Istanbul-2500 and Istanbul-5000 datasets are given in Figure 9. A more stable training/test accuracy was observed for DenseNet-169 for the Istanbul-5000 dataset (Figure 9e) than for all the other DL models used. After the 100th epoch for both the DenseNet169 and DenseNet201 architectures and the Istanbul-5000 dataset, the large fluctuations stabilized, and the test accuracies converged to the training accuracy. It should be mentioned that the model architecture and the dataset size are compatible.

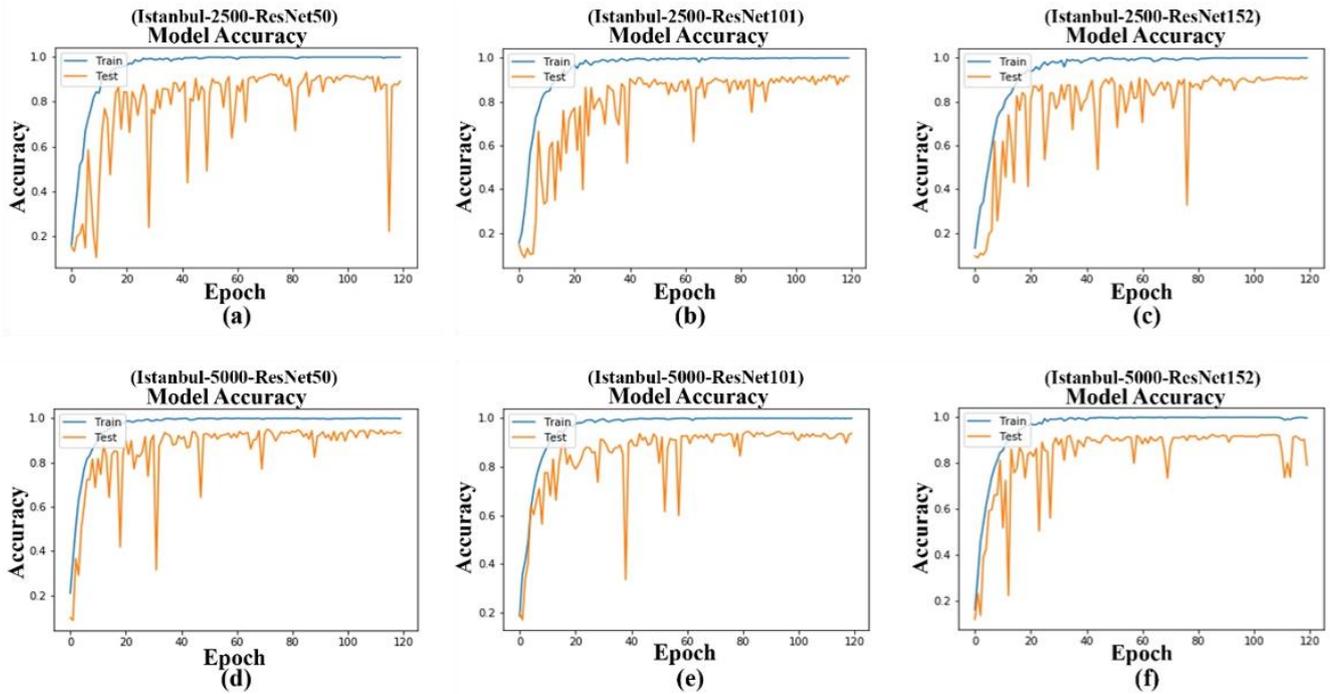


Figure 8. ResNet50, ResNet101 and ResNet152 training results for the Istanbul-2500 and Istanbul-5000 datasets

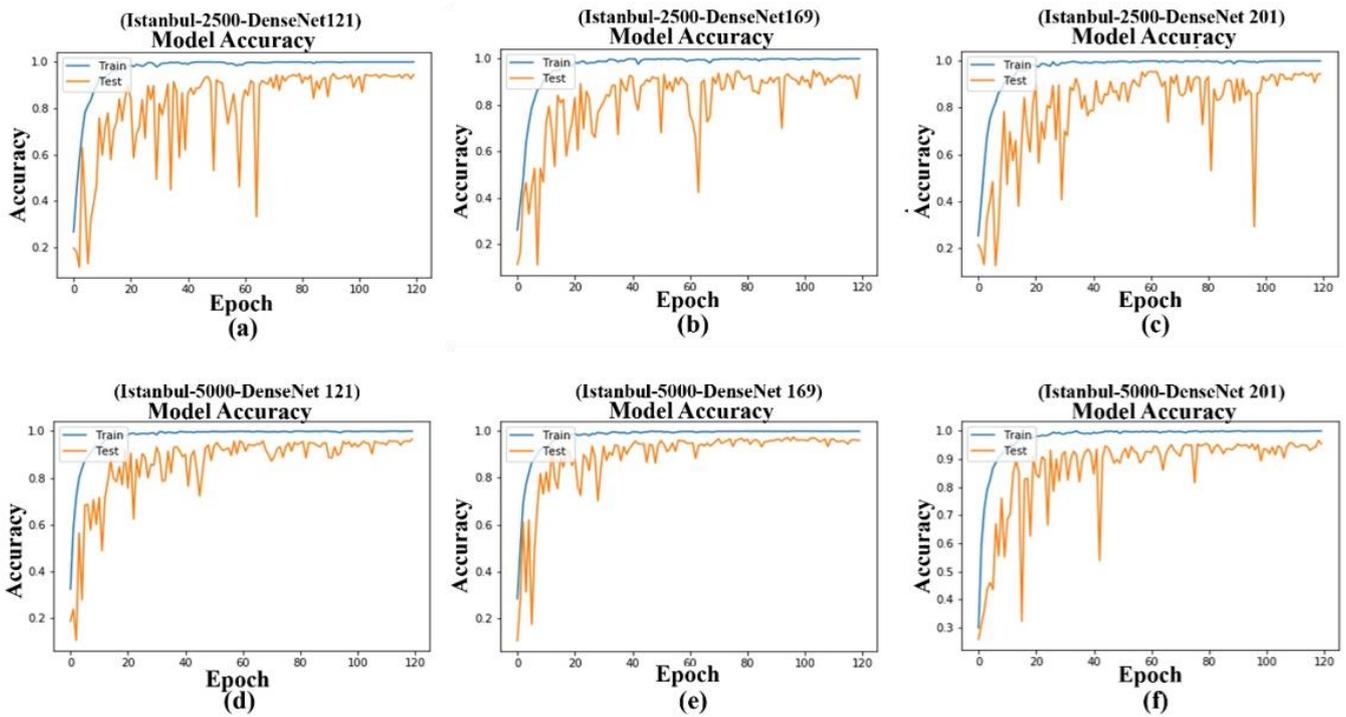


Figure 9. DenseNet121, DenseNet169 and DenseNet201 training results for the Istanbul-2500 and Istanbul-5000 datasets

In this study, precision, recall, and F1-Score metrics were used to analyze the results (Table 4). For the Istanbul-2500 dataset, the DenseNet architecture, which was able to evaluate feature maps from more layers, achieved a high level of success in all evaluation criteria for test accuracy, precision, recall, and F1-Score.

Except for ResNet-152, the mean accuracy results of the ResNets were higher than for VGG-16 and VGG-19. However, as with the VGG16 and VGG19 architectures, the accuracy results for the ResNets were found to be unsatisfactory for this study.

Table 4. Accuracy assessment results for the Istanbul-2500 and the Istanbul-5000 datasets

Deep Learning Architectures	Test Accuracy (%)		Average Precision (%)		Average Recall (%)		Average F1-Score (%)	
	Istanbul-2500	Istanbul-5000	Istanbul-2500	Istanbul-5000	Istanbul-2500	Istanbul-5000	Istanbul-2500	Istanbul-5000
VGG16	81	83.1	81.61	83.23	81	83.1	81.07	83.11
VGG19	82	83.6	82.57	83.76	82	83.6	82.07	83.61
ResNet50	90.2	93.2	91.2	93.43	90.2	93.2	90.29	93.22
ResNet101	91.6	92.4	92.32	92.46	91.6	92.4	91.68	92.41
ResNet152	90.6	81.4	91.17	85.21	90.6	81.4	90.68	81.37
DenseNet121	93.6	96.1	93.74	96.12	96.6	96.1	93.62	96.1
DenseNet169	91.8	96.3	92.23	96.35	91.8	96.3	91.88	96.3
DenseNet201	93.6	94.4	93.87	94.71	93.6	94.4	93.92	94.41

For both the Istanbul-2500 and Istanbul-5000 datasets, the superiority of the DenseNet-based architectures is conspicuous. The highest accuracy was obtained with the DenseNet-169 variant for all metrics. The test accuracy, average precision, average call, and F1-score results were calculated as 96.3%, 96.35%, 96.30%, and 96.3%, respectively. The confusion matrix for the DenseNet-169 results has been given in Table 5.

According to our results, the DenseNet-169 architecture was more successful than the other implemented DL architectures. Although the layer number of DenseNet-201 is higher than DenseNet-169, a lower accuracy was achieved compared to DenseNet-169. The reason for this situation might be related to the number of training images in the dataset.

Table 5. Confusion matrix for landmarks obtained from the DenseNet169 network

Landmarks	1	2	3	4	5	6	7	8	9	10
1	0.98	0	0.01	0	0	0	0	0	0	0.01
2	0.01	0.97	0	0	0	0	0.01	0	0	0.01
3	0.03	0.01	0.96	0	0	0	0	0	0	0
4	0	0.03	0.01	0.96	0	0	0	0	0	0
5	0	0.01	0.01	0	0.97	0	0	0.01	0	0
6	0	0	0	0.01	0	0.98	0	0	0	0.01
7	0	0	0	0	0	0	0.98	0	0.01	0.01
8	0.02	0	0.01	0.01	0.01	0	0	0.95	0	0
9	0	0	0.01	0	0	0	0.01	0	0.97	0.01
10	0	0.02	0.02	0	0	0	0.03	0.01	0.01	0.91

According to Table 5, the lowest accuracy was calculated for the Dolmabahce Clock Tower (Landmark 10) at 91%. Although the similarity between the Maiden's Tower (Landmark 1) and the Galata Tower (Landmark 3) is very high, there were no mix-ups in their recognition, as demonstrated by their results (which were 98% and 96%, respectively). Similarly, encouraging results were obtained for the Blue Mosque (Landmark 2),

Hagia Sophia (Landmark 4), and the Ortaköy Mosque (Landmark 5).

The training times for the eight DL architectures are given in Table 6. It can be seen that the training time of the ResNet-152 model for the Istanbul-5000 dataset was the longest. In contrast, ResNet-50 required minimum training time. The training time of DenseNet-169, which provided the best accuracy, was the third shortest of all the models.

Table 6. Training times of deep learning models

Istanbul-2500		Istanbul-5000	
ResNet50	0 h 38 m 54 s	ResNet50	1 h 18 m 43 s
DenseNet121	0 h 44 m 39 s	DenseNet121	1 h 29 m 45 s
DenseNet169	0 h 57 m 16 s	DenseNet169	1 h 53 m 10 s
VGG16	0 h 58 m 36 s	VGG16	1 h 57 m 31 s
DenseNet201	1 h 10 m 11s	VGG19	2 h 13 m 18 s
ResNet152	1 h 36 m 59 s	ResNet101	2 h 13 m 54 s
VGG19	1 h 6 m 59 s	DenseNet201	2 h 19 m 19 s
ResNet101	1 h 7 m 3 s	ResNet152	3 h 13 m 15 s

The Apple iOS mobile application was developed by integrating the weight file of the DenseNet169 architecture into the CoreML (.mlmodel), which can also work offline mode. The application uses either freshly taken images useful pre-existing photos on the phone. The system lists the three most probable of the ten

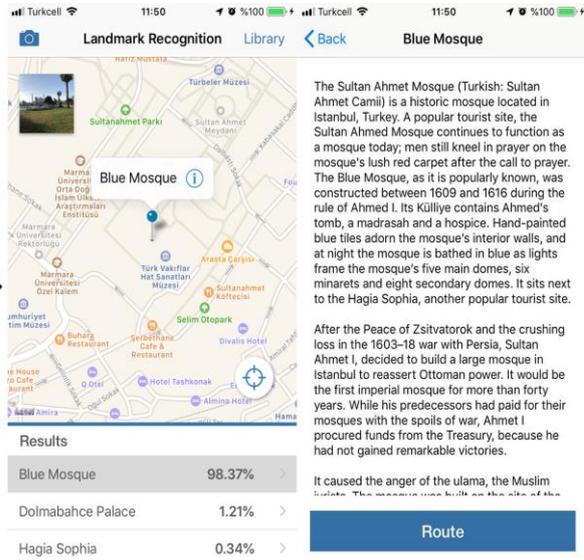


Figure 10. DL-integrated iOS-based mobile application

5. CONCLUSION

In recent years, DL architectures have been employed for different image vision and recognition studies. While some researchers have used existing DL architectures, some have developed DL architectures focusing on specific problems.

Historic landmarks, along with attribute information, are a challenging problem. It is possible to create smart systems using crowdsourcing data thanks to DL techniques. In this study, we investigated the efficiency of the DL technique for mobile historic landmark recognition. For this purpose, we explored eight different deep learning approaches to recognize ten historic landmarks of Istanbul. The highest F1-scores calculated for the DenseNets. The reasons for this could be related to the high linear and textural complexity of the selected historic objects and the number of classes. Our results show that deep learning offers a promising alternative means of recognizing historic landmarks.

The ten selected historic landmarks are unique monuments in Istanbul, and each has different textures and structures. Thus, it was possible to collect different characteristic images of the selected landmarks to generate training and test datasets. Only daylight images were included in this study. The developed framework can be extended for other historical landmarks in different cities around the world by collecting a more massive amount of data and working with more suitable computer configurations.

The proposed prototype can be used efficiently in Istanbul, which has an enormous amount of historical monuments. This system provides not only historic

included historical buildings after recognition, as well as their probabilities. The most probable structure can be selected by the user, and the system provides the location and route plan on the screen using GPS information from the phone (Figure 10).

landmark recognition but also attribute information and route planning. The developed mobile historic landmark recognition system can be implemented for other cities. As a future study, we plan to implement semantic segmentation architectures before the object recognition process as a tool for background elimination. Segmentation would prevent the learning of other objects that do not express the historic structure. Additionally, we plan to increase the number of recognizable historical buildings and to design our own deep learning architecture for historic landmark recognition purposes.

REFERENCES

- Amato, F., Moscato, V., Picariello, A., Colace, F., Santo, M.D., Schreiber, F.A., Tanca, L. (2017). Big data meets digital cultural heritage: Design and implementation of scrabs, a smart context-aware browsing assistant for cultural environments. *Journal on Computing and Cultural Heritage (JOCCH)*, 10 (1), 6.
- Brown, B. (2007). Working the problems of tourism. *Annals of Tourism Research*, 34 (2), 364-383.
- Cheng, Z., Shen, J. (2016). On very large scale test collection for landmark image search benchmarking. *Signal Processing*, 124, 13-26.
- Chollet, F. (2015). Keras: Deep learning for humans, Github. <https://github.com/keras-team/keras>
- CoreML Documentation, (2019). Converting Trained Models to Core ML, <https://developer.apple.com/documentation/core>

- ml/convertting_trained_models_to_core_ml [Accessed on 2 November 2019].
- CoreML Framework, (2019). CoreML Framework Overview, <https://developer.apple.com/documentation/coreml> [Accessed on 2 November 2019].
- Core Location Framework, (2019). Core Location Framework Overview, <https://developer.apple.com/documentation/corelocation> [Accessed on 2 November 2019].
- Deng, J., Berg, A., Satheesh, S., Su, H., Khosla, A., Fei-Fei, L. (2012). ImageNet Large Scale Visual Recognition Competition (ILSVRC2012). <http://www.image-net.org/challenges/LSVRC/2012/>.
- Doğan Y., Yakar, M. (2018). GIS and three-dimensional modeling for cultural heritages. *International Journal of Engineering and Geosciences*, 3 (2), 50–55.
- Gunawan, A.A., Surya, K., Meiliana. (2018). Brainwave classification of visual stimuli based on low cost EEG spectrogram using DenseNet. *Procedia Computer Science*, 135, 128–139.
- Hays, J., Efros, A.A. (2008). IM2GPS: Estimating geographic information from a single image. In *26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- He, K., Zhang, X., Ren, S., Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.
- He, K., Zhang, X., Ren, S., Sun, J. (2016b). Identity mappings in deep residual networks. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 630–645.
- Huang, C., Xu, H., Xie, L., Zhu, J., Xu, C., Tang, Y. (2018). Large-scale semantic web image retrieval using bimodal deep learning techniques. *Information Sciences*, 430–431, 331–348.
- Huang, F., Zhang, X., Zhao, Z., Li, Z., He, Y. (2018). Deep multi-view representation learning for social images. *Applied Soft Computing Journal*, 73, 106–118.
- Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q. (2017). Densely connected convolutional networks. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition - CVPR 2017*, pp. 2261–2269.
- IPDCT - Istanbul Provincial Directorate of Culture and Tourism, (2019a). Where I am: Maiden's Tower, <http://www.istanbulkulturturizm.gov.tr/EN-171825/maidens-tower.html> [Accessed on 12 September 2019].
- IPDCT - Istanbul Provincial Directorate of Culture and Tourism, (2019b). Where I am: Sultanahmet Mosque (Blue Mosque), <http://www.istanbulkulturturizm.gov.tr/EN-174344/sultanahmet-mosque-blue-mosque.html> [Accessed on 12 September 2019].
- IPDCT - Istanbul Provincial Directorate of Culture and Tourism, (2019c). Where I am: Galata Tower, <http://www.istanbulkulturturizm.gov.tr/EN-171079/galata-tower.html> [Accessed on 12 September 2019].
- IPDCT - Istanbul Provincial Directorate of Culture and Tourism, (2019d). Where I am: Ayasofya Museum, <http://www.istanbulkulturturizm.gov.tr/EN-171028/ayasofya-museum.html> [Accessed on 12 September 2019].
- IPDCT - Istanbul Provincial Directorate of Culture and Tourism, (2019e). Where I am: Ortakoy Mosque, <http://www.istanbulkulturturizm.gov.tr/TR-209469/ortakoy-camii.html> [Accessed on 12 September 2019].
- IPDCT - Istanbul Provincial Directorate of Culture and Tourism, (2019f). Where I am: Topkapı Palace, <http://www.istanbulkulturturizm.gov.tr/EN-174359/topkapi-palace.html> [Accessed on 12 September 2019].
- IPDCT - Istanbul Provincial Directorate of Culture and Tourism, (2019g). Where I am: Dolmabahçe Palace, <http://www.istanbulkulturturizm.gov.tr/EN-171052/dolmabahce-palace.html> [Accessed on 12 September 2019].
- IPDCT - Istanbul Provincial Directorate of Culture and Tourism, (2019h). Where I am: Dikilitas (Theodosius Sütunu), <http://www.istanbulkulturturizm.gov.tr/TR-209450/dikilitas-theodosius-sutunu.html> [Accessed on 12 September 2019].
- IPDCT - Istanbul Provincial Directorate of Culture and Tourism, (2019i). Where I am: Dolmabahce Clock Tower, <http://www.istanbulkulturturizm.gov.tr/EN-171039/dolmabahce-clock-tower.html> [Accessed on 12 September 2019].
- Jiang, B., Yang, J., Lv, Z., Tian, K., Meng, Q., Yan, Y. (2017). Internet cross-media retrieval based on deep learning. *Journal of Visual Communication and Image Representation*, 48, 356–366.
- Keras Documentation, (2019). <https://keras.io/optimizers/> [Accessed on 25 August 2019].
- Lin, M., Chen, Q., Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*.
- MapKit Framework, (2019). <https://developer.apple.com/documentation/mapkit> [Accessed on 12 September 2019].
- Maskrey M., Wang W. (2018). Using facial and text recognition. In *Pro iPhone Development with Swift 4*, pp. 285-315, Apress, Berkeley, CA.
- McGookin, D., Tahiroğlu, K., Vaitinen, T., Kytö, M., Monastero, B., Carlos Vasquez, J. (2019). Investigating tangential access for location-based digital cultural heritage applications. *International Journal of Human Computer Studies*, 122, 196–210.
- Mulazimoglu, E., Basaraner, M. (2019). User-centred design and evaluation of multimodal tourist maps. *International Journal of Engineering and Geosciences*, 4 (3), 115–128.
- Nawaz, M., Sewissy, A.A., Soliman, T.H.A. (2018). Multi-class breast cancer classification using deep learning convolutional neural network. *International Journal of Advanced Computer Science and Applications*, 9 (6), 316–332.
- Nibali, A., He, Z., Wollersheim, D. (2017). Pulmonary nodule classification with deep residual networks. *International Journal of Computer Assisted Radiology and Surgery*, 12 (10), 1799–1808.

- Parkhi, O.M., Vedaldi, A., Zisserman, A. (2015). Deep face recognition. In *Proceedings of the British Machine Vision Conference 2015*, pp. 41.1–41.12.
- Patterson J., Gibson A. (2017). Deep learning: A Practitioner's approach. *O'Reilly Media*, Sebastopol, CA, USA.
- Richards, G. (2018). Cultural tourism: A review of recent research and trends. *Journal of Hospitality and Tourism Management*, 36, 12–21.
- Rothe, R., Timofte, R., Van Gool, L. (2018). Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, 126 (2–4), 144–157.
- Schaul, T., Antonoglou, I., Silver, D. (2014). Unit tests for stochastic optimization. In *International Conference on Learning Representations*. arXiv:1312.6055.
- Shukla, P., Rautelai, B., Mittal, A. (2017). A computer vision framework for automatic description of Indian monuments. In *13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, 4–7 December, Jaipur, India.
- Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Soon, F. C., Khaw, H.Y., Chuah, J.H., Kanesan, J. (2018). Hyper-parameters optimisation of deep CNN architecture for vehicle logo recognition. *IET Intelligent Transport Systems*, 12 (8), 939–946.
- Şasi A., Yakar, M. (2018). Photogrammetric modelling of Hasbey Dar'ülhuffaz (Masjid) using an unmanned aerial vehicle. *International Journal of Engineering and Geosciences*, 3 (1), 6–11.
- Termritthikun, C., Kanprachar, S., Muneesawang, P. (2018). NU-LiteNet: Mobile landmark recognition using convolutional neural networks. *arXiv preprint arXiv:1810.01074v1*.
- Tzelepi, M., Tefas, A. (2018). Deep convolutional image retrieval: A general framework. *Signal Processing: Image Communication*, 63, 30–43.
- UNESCO, (2006). Cities Named '2010 European Capital of Culture' include World Heritage sites. <https://whc.unesco.org/en/news/248/> [Accessed on 12 September 2019].
- Weyand, T., Leibe, B. (2015). Visual landmark recognition from Internet photo collections: A large-scale evaluation. *Computer Vision and Image Understanding*, 135, 1–15.
- Xu, H., Huang, C., Wang, D. (2019). Enhancing semantic image retrieval with limited labeled examples via deep learning. *Knowledge-Based Systems*, 163, 252–266.
- Yorulmaz M., Celik, O.C. (1995). Structural analysis of the Bozdoğan (Valens) aqueduct in Istanbul. In *Arch Bridges, Ed. C.Melbourne, Thomas Telford*, London, 175–180.
- Zeiler, M.D. (2012). ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.
- Zhang, D., Han, X., Deng, C. (2018). Review on the research and practice of deep learning and reinforcement learning in smart grids. *CSEE Journal of Power and Energy Systems*, 4 (3), 362–370.
- Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A. (2018). Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40 (6), 1452–1464.
- URL1:
<https://github.com/fchollet/deep-learning-models> [Accessed 10 September 2019]
- URL2:
<https://github.com/raghakot/keras-resnet> [Accessed 10 September 2019]
- URL3:
<https://github.com/flyyufelix/DenseNet-Keras> [Accessed 10 September 2019]



© Author(s) 2020. This work is distributed under <https://creativecommons.org/licenses/by-sa/4.0/>