

Inside Iasi - Augmented Reality Application for Discovering the City of Iasi, Romania

Daria-Roxana Iacob, Adrian Iftene

“Alexandru Ioan Cuza” University of Iasi, Faculty of Computer Science
St. General Henri Mathias Berthelot 16, Iași 700483
E-mail: iacobdarioxana@gmail.com

Abstract. Augmented reality (AR) is used in more and more fields, and these are more and more diverse. From medicine, to education and games, to interior or exterior design, the applications attract various age groups, but especially young people, who are delighted by the discoveries and innovations in technology. Nowadays there are few applications that can help tourists to easily travel around cities and discover interesting information about them in an interactive way. In this paper we propose an application that is able to recognize certain touristic attractions from Iasi, Romania more exactly - the Palace of Culture, the Metropolitan Cathedral “Saint Parascheva”, “Vasile Alecsandri” National Theatre, “Mihai Eminescu” University Library and Iasi City Hall with the help of AR and with the help of convolutional neural networks (CNNs). By means of AR, once the user has arrived at the tourist destination and classified it with the help of the CNN, they will be able to have an immersive experience in which they will learn various curiosities and historical details about the place, the visiting schedule, as well as view a video that captures shots from inside the location. Moreover, every information displayed in the AR experience are voice assisted.

Keywords: Augmented reality, mobile application, convolutional neural network.

DOI: 10.37789/ijusi.2022.15.4.1

1. Introduction

The evolution of augmented reality (AR) in the last decade has led to the rethinking of many types of applications, which have been adapted to this new opportunity. While AR is often synonymous with leisure and entertainment, its profound implications span vital sectors such as medicine (Sbarcea and Iftene, 2023), education (Gușă et al., 2019), gaming (Călărășu and Iftene, 2021), culture (Constantinescu et al., 2023) and simulation and training (Rokhsaritalemi et al., 2020). This widespread applicability emphasizes the unique skill of AR: the ability to give users the opportunity

to see the surrounding reality with different eyes, which is augmented with useful information, depending on the purpose of the application. Also, in the area of tourism, several applications have been developed, which allow the identification of local attractions (Kadagidze and Ugrelidze, 2023) and cultural and historical heritage (Abdul-Jabbar and Alwehab, 2023), (Chanpum et al., 2022).

The main objective of the application proposed and described in this paper consists in discovering the main points of interest in Iasi, Romania through an interactive approach, using AR. The first component consists of a CNN capable of recognizing tourist attractions in Iasi.

To create the image classifier, we created the data by photographing different tourist attractions and also using resources found on the Internet. We trained the CNN using various models and hyperparameters, choosing for integration with the AR mobile application the model that achieved the best results, taking into account the accuracy and prediction time of an image. Another implemented component is the AR application. Within it, we have created two functionalities that consist in the detection of the tourist objective in real time with the help of the previously described classifier, with the possibility of discovering it by offering information related to the program, curiosities, frames from the inside, all being superimposed on the user's real world.

The second functionality is rendered by a text recognition and translation component in various languages, useful for foreign tourists to understand the information written on various information boards placed at a tourist attraction. Besides this, we created a Web API type application, responsible for getting various data about places in Iasi by category, communicating with the Google Maps API in order to obtain this information. Also, the application communicates with a database where data is stored for points of interest that the image classifier can recognize.

The last part represents a mobile application that has the role to integrate all the previously exposed components, in order to offer an integrated map where various tourist attractions can be viewed according to the type selected by the user, providing a route to a tourist attraction and the possibility to access the AR application in order to discover a point of interest. Moreover, it also includes a functionality that provides weather information, being able to trigger various alerts in case of extreme weather conditions.

2. Points of Interest and Associated Models

For image-based classification of the local spots from Iasi, we used CNNs. This section presents the method of obtaining the data as well as the models used for the classification.

2.1. Dataset

Some of the images were selected from the Internet especially from Instagram, Google Images, Google Maps, Google Street View and from the TripAdvisor application. The rest of the photos were taken with the help of the phone in two periods of time, December of 2022 and February of 2023. The images capture the tourist sites from different angles, being taken both close to the points of interest and from distance in order to have a wide range of ways in which they can be immortalized. For convenience, we also made videos from which we later extracted a number of frames. Some example images from the dataset for part of the points of interest are illustrated below.



Figure 1. The Palace of Culture, the National Theater and the Metropolitan Cathedral.

2.2. Trained Models for the Classification of Tourist Attractions

In order to obtain a model that classifies points of interest with high accuracy, we trained various neural network (NN) models that are presented below. The VGG16 (Simonyan and Zisserman, 2015) model is a very popular CNN, it won the ILSVRC competition in 2014 being able to classify labeled images into over a thousand categories with an accuracy of 92.7%. It was developed by researchers from Oxford University and its name comes from the acronym of the research institute Visual Geometry Group.

The model architecture consists of thirteen convolutional layers, five max

pooling and three dense layers. VGG16 takes as input an array of pixels of size (224, 224) in RGB format. For the convolutional layers, kernel matrices of size 3x3 with one move step are used, on the pixel matrices the procedure of filling with zeros was applied to preserve the dimensions of the images, and for the max pooling layer a filter of size 2x2 with step two was used. The ReLU activation function was used for the hidden layers and the SoftMax function (Dubey et al., 2022) was used in the output layer. The output layer is dense and contains five nodes, corresponding to the five classes of sights. Prior to the training process, the images go through a preprocessing step that consists of converting the images to BGR format and centralizing the pixel values to zero, the latter involves calculating the average of the data and subtracting it from each pixel of the image corresponding to each color channel. This process of averaging the data to zero helps increase the accuracy of the network because it reduces the effect of differences in light and contrast between images and focuses on their content.

The data preprocessing can be seen in Figure 2 where some of the images from the network training stage are rendered. The dataset was split into three parts: 80% for training, 10% for each of the validation and testing stages. For a first training of the VGG16 model with the aim of classifying the five tourist attractions, we set the following values for the network parameters: the learning rate of 0.0001, the Adam optimization algorithm, the cross-entropy function for the calculation of the loss function, a batch size of ten images and fifteen epochs. After training with the previously mentioned parameters, we achieved an accuracy of 99.84% on the test data, misclassifying only one instance.

Another way of training the VGG16 model was done using the technique of transfer learning (Tao et al., 2021), with a batch size of 64 images. Transfer learning involves using a pre-trained neural network on a certain task, such as image classification, to solve a similar problem. The VGG16 model was trained on the ImageNet dataset (Deng et al., 2009). To accomplish this, we modified the final layer of the network to adapt to the number of classes required for the new classification, while keeping the weights and biases of the previous layers unchanged. This training process was faster. Applying this technique to the test dataset, we achieved an accuracy of 99.69%, with the neural network making two mistakes.



Figure 2. Processed images for VGG16.

The VGG19 model (Simonyan and Zisserman, 2015) extends the VGG16 model by adding three additional convolutional layers, resulting in a larger number of parameters and a more complex structure. The VGG19 model is very deep and has been trained on millions of images from different categories. It is more accurate than its predecessor but requires more resources during the training process. Similar to the VGG16 model, the images go under preprocessing before training. The hyperparameters used in this model are the same as those in the previously presented VGG16 model, but with only ten epochs. The results obtained on the training, validation, and test datasets showed that the maximum accuracy on the test images was achieved in the eighth epoch. After that, the model overfits the training data. Subsequently, we used the same model for transfer learning with a batch size of 64 and fifteen epochs, adding five dense layers at the end. Both models made only one mistake: in the first configuration, the Palace of Culture was misidentified as the Metropolitan Cathedral, while in the second configuration, the result was the exact opposite.

The Inception V3 model (Szegedy et al., 2016) is a deep learning model based on convolutional neural networks, developed by a team at Google. It is used for image classification and consists of forty-two layers. This model is an extension of its predecessor, Inception V1, utilizing various network optimization techniques to better adapt to new training data. One improvement is the reduction of the kernel matrix size in the convolutional layers from 5×5 to 3×3 , resulting in fewer parameters to compute and requiring fewer computational resources. Through the preprocessing

function, the pixels are scaled between 1 and -1, and some examples of images from the training dataset can be found in Figure 3, with dimensions of (299, 299). For the training of the network, we used the transfer learning method by adding three dense layers and an average pooling layer at the end of the model to reduce the number of parameters. We ran fifteen epochs with a batch size of sixteen images, and the Adam optimization algorithm was used with a learning rate of 0.0001. The model reached convergence in the third epoch for the test image set, quickly adapting to the patterns present in the photographs of the training data collection.



Figure 3. Processed images Inception V3.

The custom model includes a layer for scaling the values so that each value in the pixel matrix is in the range [0, 1]. It consists of three convolutional layers with 64, 128, and 256 filters, respectively, to extract features from the images, and two max pooling layers to reduce the size of the matrices and extract relevant information. In the middle of the network, a dense layer is inserted, and at the end, a SoftMax activation function is used to map the three possible output classes. The ReLU activation function was used for the convolutional layers. The architecture of the model can be visualized in Figure 4.

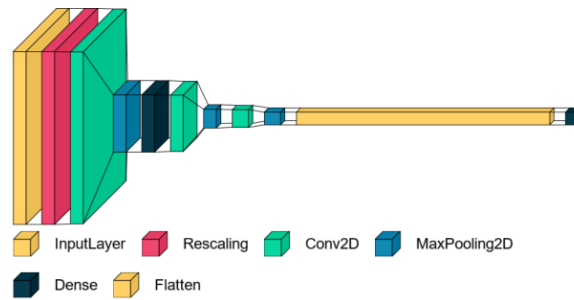


Figure 4. Custom model architecture.

Within this model, we used the technique of data augmentation. Data augmentation is used in convolutional neural networks to improve the model's performance by adding new data to the existing collection through various transformations such as rotations, adjustments in brightness and contrast, changes in colors, and horizontal flips. We utilized the Albumentations¹ library for data augmentation. The number of images resulting from the augmentation process increased from approximately 2,000 to 6,000, and a few examples of pictures can be seen in Figure 5.

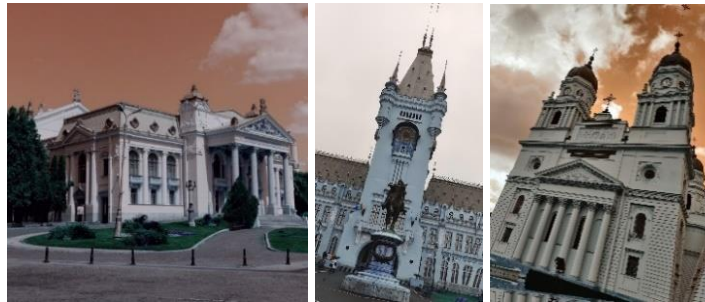


Figure 5. Examples of augmented images.

The dataset was divided into three parts: 80% for the training set, 10% for the validation set, and the remaining portion for testing. The dimensions of the images were (256, 341). During the training process, we used a batch size of 64, ten epochs, ReLU activation function for the hidden convolutional layers, and SoftMax activation function for the output layer. The model reached convergence in the sixth epoch for both the test and training datasets,

¹ <https://albumentations.ai/>

correctly classifying each image. In terms of speed, the custom model outperforms the Inception V3 model, with a runtime approximately six times faster. This makes the custom model preferable, especially for mobile app.

3. Proposed solution

From a user-friendly interaction standpoint, we have thought of structuring the tourist attractions within the application into categories: museums and theaters, parks and monuments, churches and cathedrals, libraries and universities, as well as a section featuring the most popular points of interest. Once the user selects the area they wish to explore, they can view the tourist attractions on an integrated map within the application to make everything as visual and user-friendly as possible. After reaching the desired attraction through the route provided by Google Maps, the user can take a photo of the point of interest they are in. After the system recognizes the attraction, an immersive exploration experience of the place will begin through AR.

3.1. System Architecture

The implemented system architecture is presented in Figure 6. A first component is the mobile application developed in the Angular framework, through which users can discover various tourist attractions based on their preferences and navigate to their desired destinations. They will also have access to the AR component and to additional information about the weather. To obtain the necessary data, the mobile application communicates with the backend module developed using .NET technology.

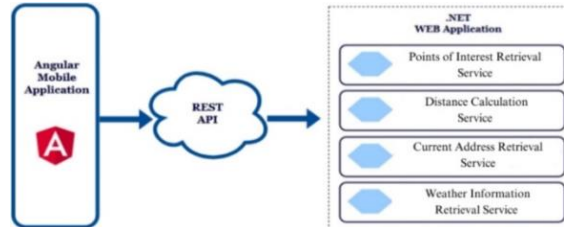


Figure 6. System Architecture.

A second component is a mobile application built in Flutter, through which users can detect a monument and explore it using augmented reality. Additionally, this component provides the capability of text recognition and

translation into various languages. The AR component is integrated in the Angular mobile application and was written in Flutter because it offers a better support for AR apps.

3.2. Inside Iasi – Backend

The backend component of the application is developed using the .NET framework and is a web application that provides various services accessible through HTTP methods such as GET or POST.

Points of Interest Retrieval Service: To obtain points of interest near a specified location using longitude coordinates, this service can be accessed with the GET verb, specifying latitude, longitude, and a keyword to filter points of interest by category. To retrieve the data, the server makes an HTTP call to the Places API. The response includes a list of found attractions, providing details such as address, name, an average user review rating, and whether it is open at the time of the request.

Distance Calculation Service: This service returns the distance in kilometers between two locations and the average travel time. It can be accessed through the GET verb, with parameters specifying the longitude coordinates of the origin and destination. To calculate the distance, the server makes a GET call to the Distance Matrix API.

Current Address Retrieval Service: In order to obtain the exact address based on longitude coordinates, this service can be accessed with the GET verb. It makes a call to the Geocoding API and returns the street name, city, and country corresponding to the received coordinates.

Weather Information Retrieval Service: This service provides weather information and can be accessed through the POST verb. It requires sending parameters in JSON format, including location, start time, end time, and fields of interest such as temperature, humidity, or precipitation probability. To obtain this weather information, the server makes an HTTP call to the Tomorrow.io API.

Image Classification Service: The image classification service can be called using the POST verb, receiving a base64-encoded image in JSON format. It predicts the name of the received tourist attraction and accesses a database to provide information about the detected location. To classify the image, the server makes a call to a Flask-based API developed in Python. It loads the custom model described earlier, runs it on the received image, and returns the predicted name of the attraction. Based on the received name, the

server queries the database and returns details about the tourist attraction, such as interesting facts, operating hours, and a link to the corresponding webpage.

3.3. Inside Iasi – Frontend

The application is available in both Romanian and English languages, and the user can switch between them using a button located in the top right corner of the screen (see Figure 7, left).

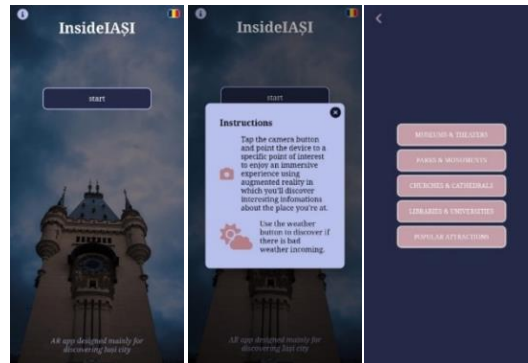


Figure 7. Start page (left), Instructions page (middle), Filtering page (right).

To implement the internationalization functionality of the application, we have created a JSON file for each available language, containing key-value pairs for the elements. Depending on the selected language, the application uses the corresponding JSON file to render the content. On the initial access to the application, the default language is English, and if the user changes it, the application will open in the previously selected language in subsequent accesses. We have used the Preferences API² to store the language preference, and the data is only deleted if the application is uninstalled. To facilitate interaction between the application and the user, there is a button located in the top left corner of the screen that can be pressed to access various information about using the application (see Figure 7, middle). The user can discover various tourist attractions in the city of Iasi based on the type of location they wish to visit. The available options include museums and

² <https://capacitorjs.com/docs/apis/preferences>

theaters, parks and monuments, churches and cathedrals, libraries and universities, or the city's most popular attractions (see Figure 7, right).

The Points of Interest Viewing Service: Once a category of tourist attractions is selected, the user can view them within the application using an integrated map powered by the Maps JavaScript API. To obtain the points of interest based on the previously chosen category, the Angular client sends a request to the server using the GET verb, specifying the longitude coordinates and a keyword specific to the category. The server will return a list of results in JSON format, specifying the location on the map, the name of the place, the average user reviews, and whether it is open at the time of the request. The received tourist attractions will be displayed on the map using markers that have corresponding icons representing their types. Three of these icons are illustrated in Figure 8.

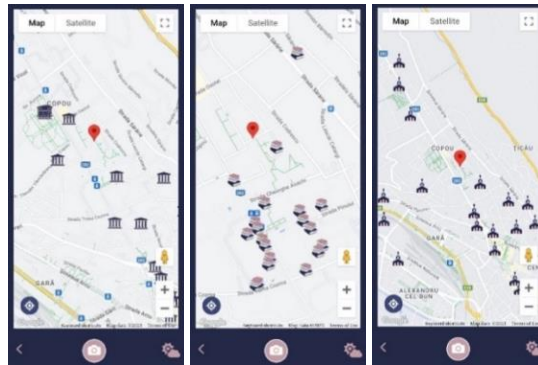


Figure 8. View museums, bookstores, churches.

Moreover, by tapping on a marker, the user can find more details about the respective attraction, such as the users review rating, its opening status, the distance from the current location to the destination, and an estimated walking time (see Figure 9, left). To determine the user's location, we used the Geolocation API³. To obtain the distance and time to the destination, the Angular client sends another GET request to the server, specifying the origin and destination coordinates as parameters. The “Visit” button provides the user with the option to choose from existing applications on their phone that allow them to obtain an exact route to the destination. When clicked, the map

³ <https://capacitorjs.com/docs/apis/geolocation>

opens in the selected application, precisely at the chosen location (as shown in Figure 9, middle).

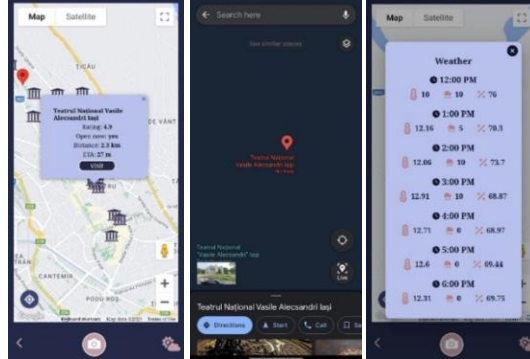


Figure 9. Objective details (left), Localization (middle), Weather (right).

Weather Service: The application also provides a component where the user can check the weather conditions for a seven-hour interval from the moment of access. This allows users to be aware of the meteorological conditions and ensure a pleasant journey. The user can view the temperature, precipitation probability and humidity percentage, as reflected in Figure 9 (right). Once the user arrives at the destination, they can return to the application and access the camera icon button located at the bottom of the screen to be redirected to the augmented reality application and begin an immersive experience where they will discover various information about the tourist attraction they are visiting. The opening of the augmented reality application is implemented using the App Launcher API, utilizing a unique identifier for the application.

3.4 Inside Iasi – Augmented Reality

This part presents the development process of the augmented reality component implemented in Flutter, which aims to detect and interactively explore a tourist attraction in the city of Iasi, Romania. Additionally, the application provides the functionality of text recognition from a photograph and translation into various internationally recognized languages. The switching between these two components is achieved through the start page, as showed in Figure 10 (left). The first button redirects the user to the location discovery component, while the next button leads to the scanning component.

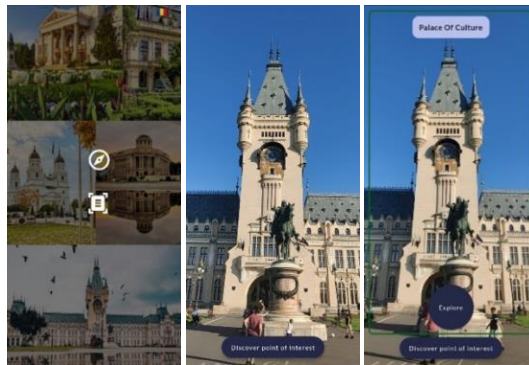


Figure 10. Example of tourist objective detection.

The application provides a primary functionality, which is the detection of the users' location using the image classifier described earlier. Once the user reaches a tourist attraction, to discover its location, they need to access the application, position the phone's camera towards the attraction, and press the detection button (Figure 10, middle). Subsequently, the attraction will be framed within a green bounding box, a label with the location's name will appear, and a button will redirect the user to the exploration functionality, as shown in Figure 10 (right). Another functionality of the application is the exploration option for a tourist attraction. It offers an immersive experience where the user can use augmented reality to discover various information about the point of interest they are visiting. This includes visiting hours, interesting facts, a video showcasing the interior of the location, and a link to the attraction's website for more details. An illustration of this feature can be seen in Figure 11.

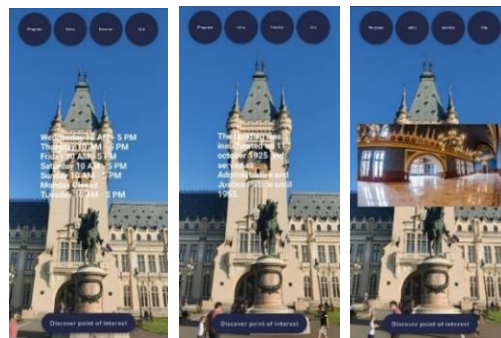


Figure 11. Exemplification of touristic objective exploration.

Furthermore, another functionality of the application is text recognition (see Figure 12). It comes with a predefined image, and below it, on the left side, there is a dropdown button where the language for translating the text from the image can be specified. The button on the right triggers a pop-up displaying the translated text. The scan button at the bottom of the screen opens the user's camera, allowing them to capture a photo containing text. They have the option to retake the photo or confirm it, as shown in Figure 12 (left). To better highlight the area for text recognition, the user can crop the image using the menu illustrated in Figure 12 (middle), which allows rotation, zooming, and cropping of the image. Subsequently, the captured photo replaces the predefined image, and the user can select the language for translating the text. The translated text appears within a bounding box, as depicted in Figure 12 (right), and can be closed by tapping outside the box. The available languages for translation include English, French, Italian, German, Romanian, Spanish, and Russian, while the language in which the text in the image is written can be any language.

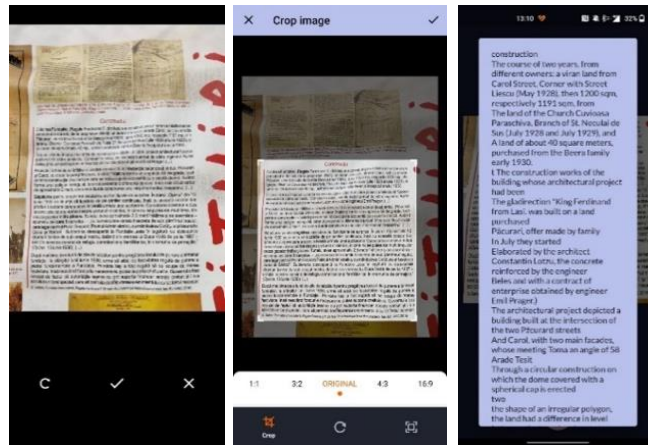


Figure 12. Example of text recognition component.

4. Usability tests

In order to test the application, we created a Google Forms survey with the purpose of collecting feedback from potential users to identify possible improvements or modifications to the application. The form consisted of a

series of questions, including multiple-choice and open-ended questions, and was addressed to a diverse group of individuals, with or without technical background. Before testing the application, users were presented with two possible usage scenarios. The first scenario was designed for users who wanted to have a complete experience with the application. The second scenario involved the use of the augmented reality feature and was intended for users exploring the city of Iasi who encountered a tourist attraction of interest and wanted to learn more about it.

The form was completed by 10 individuals ranging in age from 19 to 57, including 6 male and 4 female participants. The group consisted of individuals with technical and non-technical backgrounds, with three participants having prior experience with AR applications and the others being first-time users of this technology. The diversity of the participants aimed to include a wide range of potential application users in order to gather diverse opinions and suggestions for improvement.

Participants were provided with instructions on how to use the application, then they tested the application by following one of the usage scenarios described earlier. Finally, they completed the feedback form. Figure 13 presents the evaluation of the usability instructions.

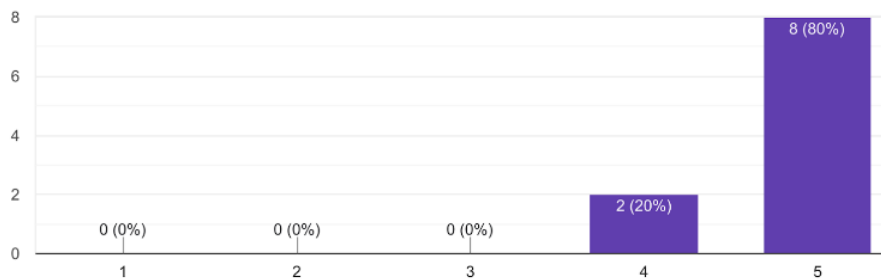


Figure 13. The clarity of the instructions received for using the application.

The participants were asked to rate various aspects of the application on a scale of 1 to 5, including design and ease of use. The scores for each aspect are displayed in Figure 14.

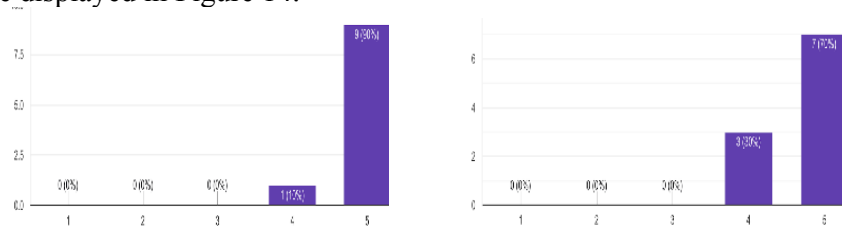


Figure 14. Ease of use (left), Design (right).

Figure 15 shows the component of the application that was most appreciated by the participants. The majority vote for the augmented reality component indicates that the goal of introducing novelty and dynamism into the application has been achieved.

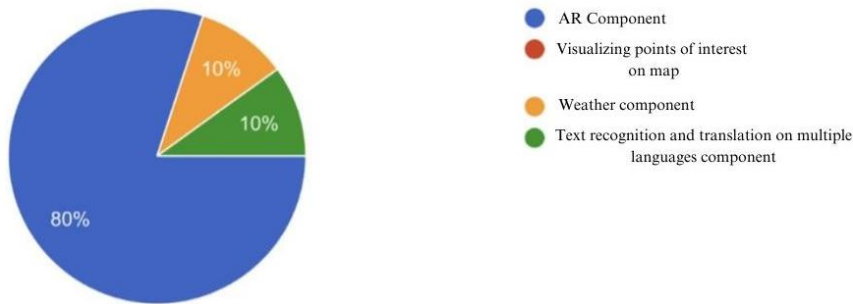


Figure 15. The most popular component of the application.

Afterwards, participants were asked to express their opinion regarding the usefulness of the application and provide a final rating for the application, as depicted in Figure 16 (left). All participants mentioned that they would recommend the application to a friend.

Finally, participants were asked about any additional elements that could be added to improve the application. One participant mentioned that it would be useful to include information about transportation options to reach a tourist attraction, which could be accessed by the user when redirected to Google Maps, where this feature is integrated.

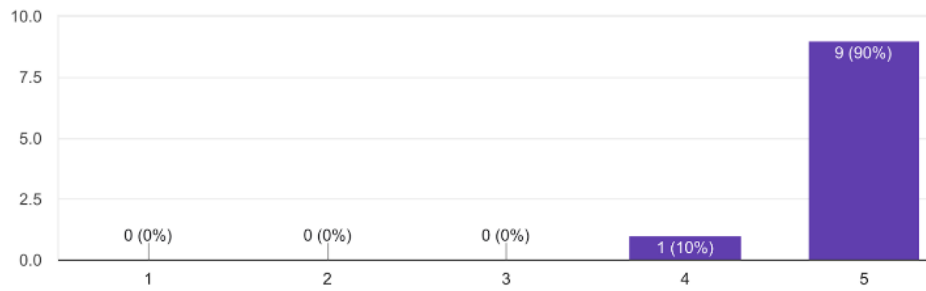


Figure 16. The utility of the application.

Another improvement suggested was the creation of a user account where users can save useful information such as favorite locations, previous experiences, and reviews that can be viewed by other users. The implementation of user accounts would require the development of a registration and authentication mechanism. Additionally, the possibility of creating a guided tour with multiple tourist attractions was also proposed. Other participants mentioned that they had no additional suggestions and were satisfied with the current state of the application.

5. Conclusion

Inside Iasi - The Augmented Reality Application for Discovering Iasi, Romania is available in Romanian and English, and offers an interactive method of exploring tourist attractions using augmented reality. It provides the ability to detect points of interest and access interesting information about them in an immersive experience. Additionally, it offers the functionality of visualizing attractions through an integrated map within the application. It also includes other useful features such as weather information, text recognition, and translation into various languages.

In the future, the image classifier could be expanded to predict a wider range of points of interest, requiring additions to the database containing information about them. Furthermore, the option of creating a user account could be introduced, allowing users to save various information and implementing a recommendation system for tourist attractions based on their previous experiences.

References

- Abdul-Jabbar, S., and Alwehab, A. *Augmented Reality's Role in Highlighting Historical Heritage/Kirkuk Citadel*. Journal of Engineering, 29, (2023), 104-122. [10.31026/j.eng.2023.12.07](https://doi.org/10.31026/j.eng.2023.12.07).
- Calarașu, T.C., and Iftene, A. *Virtual Reality FantasyShooter*. In Proceedings of the Workshop on Intelligent Information Systems WIIS2021, (2021), 76-84.
- Chanpum, P., Maneechotpeeti, P., Nuallaong, T., and Laksanaboonsong, J. *Augmented Reality Mobile Application for Thailand's Heritage Tourism Promotion*. In 5th International Conference on Business and Information Management (ICBIM), (2022).
- Constantinescu, C.G., Stamate, V., Filimon, D., and Iftene, A. *Book Reckon – The use of Virtual Reality in the Creation of Libraries of the Future*. In International Conference on INnovations in Intelligent SysTems and Applications (IEEE INISTA), (2023). <https://ieeexplore.ieee.org/document/10310470>
- Deng, J., Dong, W., Socher, R., Li, L. -J., Li, K., and Fei-Fei, L. *ImageNet: A large-scale hierarchical image database*. 2009 IEEE Conference on Computer Vision and Pattern Recognition, (2009), 248-255. <https://ieeexplore.ieee.org/document/5206848>
- Dubey, S.R., Singh, S. K., and Chaudhuri, B. B. *Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark*, arXiv: 2109.14545 (2022), 4.
- Gușă, D.R., Iftene, A., and Gifu, D. *Solar System Explorer*. In 5th Proceedings of the Conference on Mathematical Foundations of Informatics, (2019), 295-304.
- Kadagidze, L., and Ugrelidze, E. *Virtual and Augmented Reality in Adventure Tourism: A Review of Applications and Future Prospects*. In VIII International New York Conference on Evolving Trends in Interdisciplinary Research and Practices, (2023).
- Rokhsaritalemi, S., Sadeghi-Niaraki, A., and Choi, S. M. *A Review on Mixed Reality: Current Trends, Challenges and Prospects*. Applied Sciences, 10 (2), (2020). <https://doi.org/10.3390/app10020636>
- Sbarcea, S.V., and Iftene, A. *Treating Acrophobia with Virtual Reality*. In Proceedings of 20th International Conference on Human- Computer Interaction (RoCHI 2023), (2023).
- Simonyan, K., Zisserman, A. *Very deep convolutional networks for large-scale image recognition*. In 3rd International Conference on Learning Representations (ICLR) (2015), 1-14.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. *Rethinking the Inception Architecture for Computer Vision*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2016), 2818-2826. <https://ieeexplore.ieee.org/document/7780677>
- Tao, J., Gu, Y., Sun, J., Bie, Y., and Wang, H. *Research on VGG16 Convolutional Neural Network Feature Classification Algorithm based on Transfer Learning 2021*. In 2nd China International SAR Symposium (CISS), Shanghai, China (2021), 1-3. <https://ieeexplore.ieee.org/document/9652277>.