Tour Scan: A Mobile Application For Cultural Artifact Recognition

Mohamed Mamdouh, Mostafa Kamel, Saber Sayed, Kholoud Abdelmaugoud, Dina Sadiq, Mai Ismail, Esraa Hossien

Faculty of Computer and Information, [Egyptian E-Learning University].

Abstract

Egypt is home to a wealth of ancient cultural heritage that continues to draw the attention of global travelers and history enthusiasts. This paper presents **TourScan**, a novel mobile application that applies deep learning techniques to support the recognition and understanding of museum artifacts in real time. Leveraging a compact convolutional neural network architecture optimized for mobile devices, **TourScan** offers interactive features such as multilingual descriptions, voice output, and AI-powered guidance. The system is validated through extensive testing on a dedicated artifact image dataset, illustrating its capability to enhance accessibility and visitor engagement in museum contexts. The findings highlight the potential of integrating AI into cultural tourism for more immersive and inclusive experiences.

Keywords: Mobile Application, Cultural Heritage, Egyptian Artifacts, Image Recognition, Convolutional Neural Network (CNN), MobileNet, Machine Learning, Multilingual Support.

1.Introduction

Museums serve as fundamental institutions for the preservation and dissemination of human civilization and history. Despite their cultural significance, many museum visitors—particularly international travelers and individuals with accessibility needs—often encounter barriers in understanding the context and details of displayed artifacts. Conventional interpretive tools such as printed guides or static labels are typically insufficient in delivering engaging, personalized, and inclusive experiences.

As one of the world's most iconic repositories of ancient history, the Egyptian Museum contains vast collections from various periods of Egyptian civilization. However, the lack of real-time, interactive guidance can make navigation and comprehension difficult. To address this, we introduce *TourScan*, a mobile application that leverages real-time image recognition using a Convolutional Neural Network (CNN), designed specifically for efficient deployment on mobile platforms.

2. Related Work

Artificial intelligence has seen increasing integration into cultural tourism technologies in recent years. Wang et al. [2] introduced a smart guide application that generates cultural recommendations based on users' location and preferences. Similarly, Kowalczyk et al. [3] employed IoT-driven systems to adapt tourist suggestions dynamically, considering environmental factors such as weather and time. Inclusive tourism models have also emerged, with Han et al. [4] incorporating facial recognition and speech synthesis to improve accessibility for individuals with disabilities.

Convolutional Neural Networks (CNNs) have proven effective across a wide range of visual recognition tasks. For example, Chang et al. [1] applied the Xception CNN model to detect environmental anomalies on beaches, achieving notable performance. However, these applications are often focused on outdoor or ecological data rather than cultural or museum-based content.

In contrast, TourScan addresses the niche domain of indoor artifact classification within museums. While prior attempts used Haar cascades and artificial neural networks (ANN) to detect and recognize faces of historical figures [14–16], our system scales this concept by employing MobileNet for broader classification of over 25 distinct pharaonic artifacts.

Figure 1. Recognizing historical figures.

Unlike previous solutions, TourScan prioritizes mobile optimization, bilingual interaction, and accessibility features to deliver a real-time, context-aware user experience inside museum environments. While prior studies such as [2], [3], and [4] have explored AI applications in cultural tourism—including context-aware recommendation systems and inclusive guidance using speech or face recognition—these solutions often remain conceptual or lack practical integration in mobile environments. In contrast, *TourScan* delivers a complete, deployable solution that combines a lightweight CNN (MobileNet V2) for real-time artifact recognition with multilingual user interaction, speech output, and a chatbot interface tailored to museum settings. Notably, the system operates entirely on-device, ensuring offline accessibility—an essential feature for tourists in heritage sites with limited connectivity. Additionally, the model was trained on a custom dataset of Egyptian museum artifacts, a domain that remains underrepresented in current literature. This practical implementation, language support, and domain-specific design distinguish *TourScan* from existing works and contribute a novel, user-centric advancement in AI-powered cultural heritage applications.

3. Methodology

To ensure efficient on-device deployment, the proposed system adopts MobileNet V2, a convolutional neural network known for its lightweight architecture and suitability for mobile environments. Unlike larger networks such as VGG16 or ResNet50, MobileNet V2 leverages depthwise separable convolutions to significantly reduce the number of parameters and computational cost while maintaining competitive accuracy.

A. Dataset

A custom dataset was created comprising 10,333 annotated images of Egyptian artifacts, categorized into 25 distinct classes. The data was divided into 9,822 training images and 511 testing samples, ensuring balanced representation across classes.

B. Preprocessing

Prior to training, images were resized to 224×224 pixels and normalized to ensure consistency. Data augmentation techniques—including horizontal flips, random rotations, and zoom operations—were applied to improve the model's robustness and generalization.

C. Model Training

The model was trained using the TensorFlow framework, with the Adam optimizer set at a learning rate of 1e-4. Training was performed for 20 epochs with the inclusion of early stopping and model checkpointing mechanisms to prevent overfitting and preserve the best-performing model.

Figure 2. CNN (MobileNet) training log across 20 epochs.

To better understand how the CNN responds to artifact inputs, we visualize the learned feature maps across several convolutional layers.

Figure 3. CNN feature map visualization for artifact input.

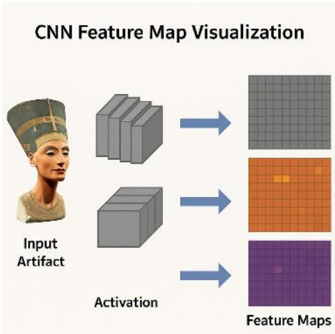
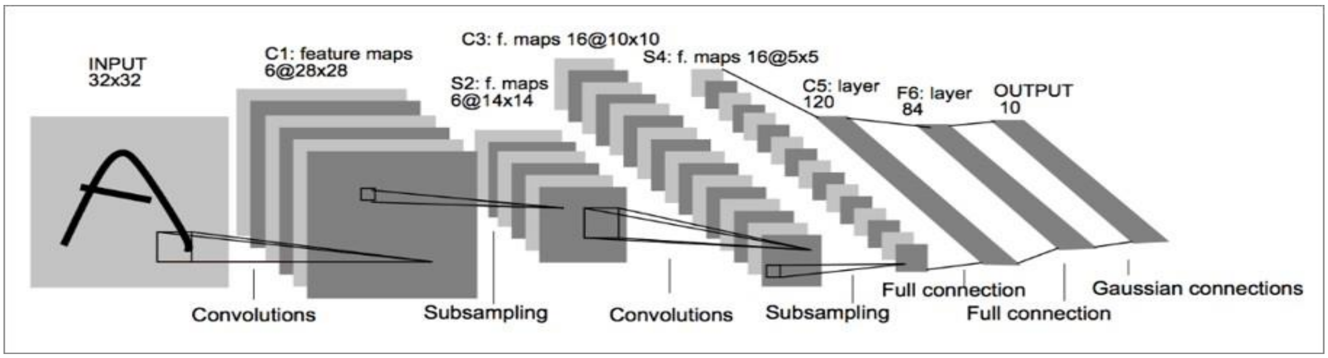


Figure 3. CNN feature map visualization for artifact input.

To illustrate this distinction, we include a classical CNN architecture (e.g., LeNet), which uses full convolutions, subsampling, and fully connected layers.



Source: LeCun et al. (1998)

Figure 4. Classical CNN architecture (LeNet-style) showing convolution, pooling, and fully connected layers.

4. Evaluation and Results

To assess the effectiveness of the proposed model, MobileNet V2 was trained and evaluated using the custom artifact dataset. The model demonstrated excellent generalization performance, reaching a validation accuracy close to 98% over 20 epochs. The training process exhibited stable convergence behavior, as illustrated by the model's accuracy trend.

For comparative evaluation, classical machine learning algorithms—namely k-Nearest Neighbors (KNN) and Support Vector Machines (SVM)—were also implemented using the same dataset. The KNN model, augmented with PCA-based dimensionality reduction, achieved an accuracy of approximately 52.96%, whereas the SVM classifier with a linear kernel reached a lower performance of 45.30%. These results highlight the significant performance advantage of deep CNN-based architectures for artifact classification tasks in mobile environments.

following:

Figure 5 shows the accuracy trend over the training epochs, highlighting the model's learning progression and convergence behavior.

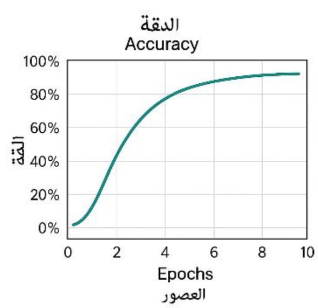


Figure 5. Accuracy curve of the CNN model across training epochs with bilingual labeling.

To benchmark performance, we compared CNN with classical models such as KNN and SVM on the same dataset

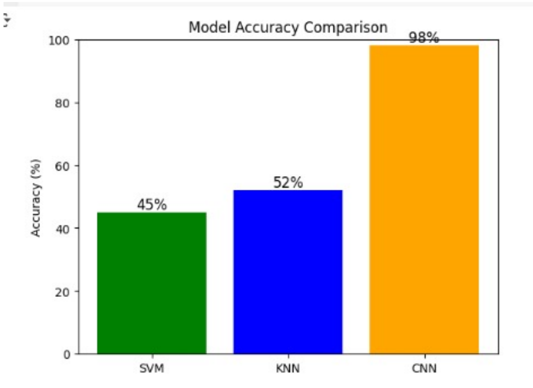


Figure 6. Bar chart comparing CNN, KNN, and SVM model performance on artifact classification.

We also implemented a baseline KNN model using PCA for dimensionality reduction, which yielded significantly lower performance.

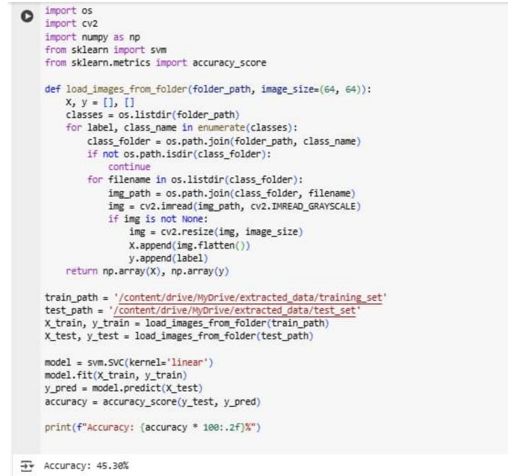
To benchmark performance, we compared CNN with classical models such as KNN and SVM on the same dataset

```
from sklearn.decomposition import PCA
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import numpy as np
import os
import cv2

def load_images_from_folder(folder_path, image_size=(64, 64)):
    X, y = [], []
    classes = os.listdir(folder_path)
    for label, class_name in enumerate(classes):
        class_folder = os.path.join(folder_path, class_name)
        if not os.path.isdir(class_folder):
            continue
        for filename in os.listdir(class_folder):
            img_path = os.path.join(class_folder, filename)
            img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
            if img is not None:
                img = cv2.resize(img, image_size)
                X.append(img.flatten())
                y.append(label)
    return np.array(X), np.array(y)

train_path = '/content/drive/MyDrive/extracted_data/training_set'
test_path = '/content/drive/MyDrive/extracted_data/test_set'

X_train, y_train = load_images_from_folder(train_path)
X_test, y_test = load_images_from_folder(test_path)

pca = PCA(n_components=50)
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train_pca, y_train)

y_pred = knn.predict(X_test_pca)

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}%")
```

```
Accuracy: 52.96%
```

Figure 7. KNN model code and output showing 52.96% classification accuracy.

An SVM model with a linear kernel was also tested, but its accuracy was lower than both KNN and CNN.



```
import os
import cv2
import numpy as np
from sklearn import svm
from sklearn.metrics import accuracy_score

def load_images_from_folder(folder_path, image_size=(64, 64)):
    X, y = [], []
    classes = os.listdir(folder_path)
    for label, class_name in enumerate(classes):
        class_folder = os.path.join(folder_path, class_name)
        if not os.path.isdir(class_folder):
            continue
        for filename in os.listdir(class_folder):
            img_path = os.path.join(class_folder, filename)
            img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
            if img is not None:
                img = cv2.resize(img, image_size)
                X.append(img.flatten())
                y.append(label)
    return np.array(X), np.array(y)

train_path = '/content/drive/MyDrive/extracted_data/training_set'
test_path = '/content/drive/MyDrive/extracted_data/test_set'
X_train, y_train = load_images_from_folder(train_path)
X_test, y_test = load_images_from_folder(test_path)

model = svm.SVC(kernel='linear')
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)

print(f"Accuracy: {accuracy * 100:.2f}%")
```

```
Accuracy: 45.30%
```

Figure 8. SVM model code and output showing 45.30% classification accuracy.

5. System Integration

TourScan was designed with an emphasis on user accessibility and seamless interaction. The application guides users through a streamlined process, starting with capturing an image of the artifact and culminating in a multilingual, descriptive response that includes historical background and translated content.

The system architecture integrates multiple components to enable real-time functionality: a CNN-based recognition engine, a Firebase cloud backend for structured artifact data storage and retrieval, a multilingual text-to-speech (TTS) engine for audio feedback, and an AI-driven chatbot that provides contextual assistance in both Arabic and English.

The intuitive interface and smooth integration of these technologies ensure a comprehensive and accessible user experience within museum environments.

Figure 9. TourScan app wireframe showing the flow from scanning an artifact to displaying translated results.

Figure 10 presents the system architecture, showing how image recognition, data retrieval, and voice interaction are integrated within TourScan.

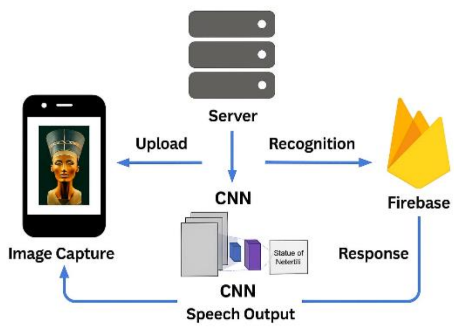


Figure 10. TourScan system architecture integrating CNN, Firebase backend, chatbot, and multilingual TTS engine.

An integrated AI-based chatbot enables users to ask questions and receive instant information about artifacts.

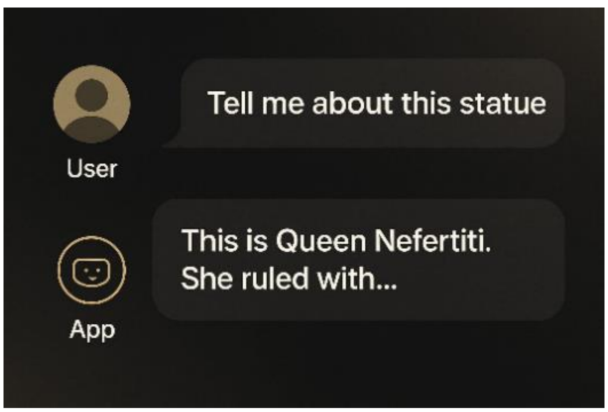


Figure 11. TourScan chatbot interface responding to a user inquiry with historical details about the artifact.

6. Discussion

The deployment of MobileNet within a mobile-based recognition system proved effective in delivering rapid and accurate artifact classification suitable for on-site museum usage. The use of a domain-specific dataset comprising Egyptian cultural artifacts was a key factor in achieving strong model performance. Comparative evaluations confirmed the advantage of deep learning techniques, particularly convolutional neural networks, over classical machine learning approaches in visual pattern recognition tasks within the cultural heritage domain.

Additionally, the incorporation of multimodal interaction—combining text, audio, and chatbot responses—substantially enhances user accessibility, particularly for those facing language or visual challenges. Despite these strengths, some limitations persist, notably the relatively constrained dataset and the lack of features such as real-time object localization or immersive augmented reality (AR) capabilities. Future iterations of TourScan may benefit from enhancements such as pose estimation, 3D modeling, and broader dataset expansion to offer deeper, more interactive museum experiences.

7.Conclusion

This study presents **TourScan** as an effective and scalable solution for enhancing cultural tourism through real-time artifact recognition on mobile platforms. By leveraging the MobileNet architecture, the system achieves a balance between computational efficiency and classification accuracy, making it suitable for deployment in museum environments. TourScan also supports multilingual and multimodal user interaction, enabling broader accessibility.
Looking forward, the project can be extended by incorporating augmented reality, expanding artifact categories, and scaling the system to include multiple heritage sites across Egypt. These enhancements aim to provide a deeper, more immersive cultural exploration experience for diverse audiences.

8. References

[1] M. Chang, Y. Y. Xing, Q. Y. Zhang, S. J. Han, and M. Kim, "A CNN Image Classification Analysis for 'Clean-Coast Detector'," Journal of Distribution Science, vol. 18, no. 1, pp. 15–26, 2020.
[2] H. Wang et al., "Smart Travel Guide Application Based on Location and Cultural Recommendation," Procedia Computer Science, vol. 124, pp. 129–136, 2017.
[3] M. Kowalczyk et al., "Deep Learning-Based Smart Tour Guide for IoT Applications," Neural Computing and Applications, vol. 33, pp. 13751–13764, 2021. [4] Y. Han et al., "Intelligent Multimodal Tourist Guidance System for Inclusive Travel," Journal of Ambient Intelligence and Humanized Computing, 2024.
[5] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in Proc. CVPR, 2017.

[6] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv:1704.04861, 2017.
[7] Y. LeCun et al., "Gradient-Based Learning Applied to Document Recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
[8] Alnaggar M, Handosa M, Medhat T, Z Rashad M. Thyroid Disease Multi-class Classification based on Optimized Gradient Boosting Model. Egyptian Journal of Artificial Intelligence. 2023 Apr 1;2(1):1-4.
[9] Alnaggar M, Handosa M, Medhat T, Rashad MZ. An IoTbased Framework for Detecting Heart Conditions using Machine Learning. International Journal of Advanced Computer Science and Applications. 2023;14(4).
[10] Alhussan AA, Talaat FM, El-kenawy ES, Abdelhamid AA, Ibrahim A, Khafaga DS, Alnaggar M. Facial Expression Recognition Model Depending on Optimized Support Vector Machine. Computers, Materials & Continua. 2023 Jul 1;76(1).
[11] Siam AI, Gamel SA, Talaat FM. Automatic stress detection in car drivers based on non-invasive physiological signals using machine learning techniques. Neural Computing and Applications. 2023 Jun;35(17):12891-904.
[12] Rohit Kundu,"YOLO: Algorithm for Object Detection Explained [+Examples]", in 2023,
[13] Alharbey R, Dessouky MM, Sedik A, Siam AI, Elaskily MA. Fatigue state detection for tired persons in presence of driving periods. IEEE Access. 2022 Jun 22;10:79403-18.
[14] Alnaggar M, Siam AI, Handosa M, Medhat T, Rashad MZ. Video-based real-time monitoring for heart rate and respiration rate. Expert Systems with Applications. 2023 Sep 1;225:120135.
[15] Morris Franken and Jan C. van Gemert, “Automatic Egyptian Hieroglyph Recognition by Retrieving Images as Texts”, in 2013, Intelligent Systems Lab Amsterdam (ISLA), University of Amsterdam Science Park 904 Amsterdam, The Netherlands.
[16] Basem H. A. Ahmed, Ayman S. Ghabayen,” Arabic Automatic Speech Recognition Enhancement”, in 2017, Palestinian International Conference on Information and Communication Technology
[17] Ankit Pandey, Vaibhav Vashist, Prateek Tiwari , Sunil Sikka, Priyanka Makkar, “Smart Voice Based Virtual Personal Assistants with Artificial Intelligence”, in 2020 , Amity School of Engineering and Technology, Haryana, India.
[18] Aradhya Saini, Tanu Gupta, Rajat Kumar, Akshay Kumar Gupta,Monika Panwar, Ankush Mittal, “Image based Indian Monument Recognition using Convoluted Neural”, in 2017, 2017 International Conference on Big Data, IoT and Data Science (BID).
[19] Aman Preet Gulati, “Hand landmarks detection on an image using Mediapipe”, in 2022, Analytics Vidhya 13. Ahmed K Elnagar , Abdelkader Mohammed Sghaier Derbali , ”The Importance of tourism contributions in Egyptian economy”, in September 2020, https://www.researchgate.net/publication/3 4401396 2_The_importance_of_tourism_contribution s_in_E gyptian_economy
[20] IBM, "What is Computer vision?", What is Computer Vision? | IBM. (n.d.). https://www.ibm.com/topics/computervision.
[21]Sipeed Wiki, "Introduction of MaixPy", in 2018, https://maixduino.sipeed.com/en/get_st arted/get_ha rdware.html.
[22] kaspersky, "What is Facial Recognition – Definition and Explanation", https://www.kaspersky.com/resourcecenter/defi nitions/w hat-is-facial-recognition
[23] Ahmed Hisham Misbah, "Can computer Vision recognize an ancient statue face?", in 2019, Research department at Artificial intelligence technology centerMisr university for Science and technology
[24]Ali Tarhini, "Face Recognition: An Introduction", in 2010, https://alitarhini.wordpress.com/2010/12/05/fac erecogniti on-an-introduction/
[25] Firas Husham Al-Mukhtar,Mustafa Zuhaer Nayef AlDabagh, "Real-Time Face Recognition System Using KPCA, LBP and Support Vector Machine" in 2017 ,International Journal of Advanced Engineering Research and Science (IJAERS)
[26] Benjamin Pryke,"How to Use Jupyter Notebook: A Beginner’s Tutorial", in 2020,

dataquest blog, https://www.dataquest.io/blog/jupyter-notebooktutorial/
[27] https://pypi.org/project/face-recognition/
[28] MaixHub, “AI online training and sharing platform” https://maixhub.com/home
[29] Maixduino example sketch "selfie" not working, in 2020, https://forum.arduino.cc/t/maixduinoexamplesketch-selfie-not-working/667685 28. MaixPy_scripts on GitHub, in 2019, https://github.com/sipeed/MaixPy_scripts
[30] GitHub Repository – TourScan Project: https://github.com/iMohamedMamdouh/TourScan_v2 8hc46kZ9pa47y4fSMfCdleH
[31] Google Colab – TourScan Model Training: https://colab.research.google.com/drive/1WGmf xmy_- 8hc46kZ9pa47y4fSMfCdleH [32] Dataset Repository – Artifact Images Collection: https://drive.google.com/drive/folders/16Zt2lRvS TzqRNdjU YfJkyFisQcwj31JM
[32] Dataset Repository – Artifact Images Collection: https://drive.google.com/drive/folders/16Zt2lRvS TzqRNdjU YfJkyFisQcwj31JM