

Faculdade de Engenharia da Universidade do Porto



**DeepRevive: Deep Learning-Based Image Analysis
for Cultural Heritage Preservation, Restoration
and Accessibility**

Miguel José de Lacerda e Megre Jordão Duarte

Dissertação realizada no âmbito do
Mestrado em Engenharia Eletrotécnica e de Computadores

Orientadora: Professora Maria Teresa Magalhães da Silva Pinto Andrade

<29 de janeiro de 2024>

“A automatização já não é apenas um problema para quem trabalha na indústria de fabrico. O trabalho físico foi substituído por robôs, o trabalho mental será substituído pela IA.” - Andrew Yang.

Resumo

Os bens do património cultural têm um enorme valor histórico, artístico e cultural, e a sua preservação, restauro e acessibilidade são cruciais para as gerações futuras.

Nos dias que correm, em que tudo é feito digitalmente e gerido por algoritmos, a relevância e importância da Inteligência Artificial não precisa de ser motivada e, com grandes avanços na área, existe a oportunidade de aproveitar estas tecnologias para melhor documentar estes bens, facilitando tanto a manutenção como a acessibilidade ao património cultural.

Esta dissertação foca-se no desenvolvimento e avaliação de uma abordagem baseada em inteligência artificial para a análise de imagens, com o objetivo de contribuir para o avanço na área da acessibilidade e preservação do património cultural, utilizando tecnologias existentes e explorando possibilidades de novas soluções.

“Automation is no longer just a problem for those working in manufacturing. Physical labor was replaced by robots, mental labor will be replaced by AI.” - Andrew Yang.

Abstract

Cultural heritage assets have great historical, artistic, and cultural value, and their preservation, restoration and accessibility are crucial for future generations.

In current days, where everything is done digitally and run by algorithms, the relevance and importance of Machine Learning, Neural Networks and Computer Vision hardly needs motivation, and with great advances, there is an opportunity to leverage these technologies to better document the assets, facilitating both maintenance and accessibility to cultural heritage.

This dissertation focuses on the development and evaluation of an AI-based approach to image analysis, with the aim of contributing to the advancement in the field of Cultural Heritage accessibility and preservation, using existing technologies and exploring the possibilities of new solutions.

Página em branco

Index

Resumo	v
Abstract.....	vii
Index	ix
List of figures	x
List of tables	xi
List of abbreviations	xii
1 Introduction	1
1.1 - Context and Motivation	1
1.2 - Objectives and Contributions	3
1.3 - Organization and Content	3
2 State of the Art	5
2.1 - Machine Learning	6
2.2 - Computer Vision	8
2.3 - AI for Cultural Heritage	9
2.4 - Conclusion	15
3 Methodology	17
3.1 - Designing a convolutional neural network	17
3.2 - Object detector	30
3.3 - Conclusion	31
4 Results	33
5 Conclusion and future work	38
References	40

List of figures

Figure 1: Illustration of a simple multilayer perceptron	10
Figure 2: Illustration of a simple CNN	11
Figure 3: Sample image of the classification dataset (bell tower).....	18
Figure 4: Sample image of the style identification dataset (gothic)	19
Figure 5: Example of images generated through data augmentation.....	21
Figure 6: Softmax activation function	25
Figure 7: ReLu activation function	26
Figure 8: Image pyramid	31
Figure 9: Result of the object detector.....	33
Figure 10: Result of architectural style classification	34
Figure 11: Model summary	34
Figure 12: Final accuracy and loss.....	37

List of tables

Table 1: Element classification dataset	Table 2: Style identification dataset.....	19
Table 3: Weights of the elements classification dataset		29
Table 4: Additional metrics		35
Table 5: Confusion Matrix		36

List of abbreviations

List of abbreviations (in alphabetical order)

ADAM	Adaptive Moment Estimation
AdaGrad	Adaptive Gradient Algorithm
AI	Artificial Intelligence
ANN	Artificial Neural Network
CH	Cultural Heritage
CNN	Convolutional Neural Network
CV	Computer Vision
DL	Deep Learning
GAP	Global Average Pooling
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
ICH	Intangible Cultural Heritage
ML	Machine Learning
R-CNN	Region Bases Convolutional Neural Network
ReLU	Rectified Linear Unit
ResNet	Residual Neural Network
RMSprop	Root Mean Squared Propagation
RNN	Recurrent Neural Network
ROI	Region of Interest
TCH	Tangible Cultural Heritage
UNESCO	United Nations Educational, Scientific and Cultural Organization

1 Introduction

The rapid advancements in Artificial Intelligence (AI), in particular Machine Learning (ML) technologies in the past decade have created significant opportunities in various fields. One key area that stands to greatly benefit from this is the field of cultural heritage. Cultural heritage preservation and accessibility is a multidimensional task that involves detecting, analyzing, maintaining, and document historical and cultural artifacts, elements, and sites.

A growing interest lies in the convergence of AI technologies, including Machine Learning and Computer Vision (CV), with the needs and processes of this field. This has led to distinct research paths, where some are focused on providing new methodologies, and others are focused on improving existing models or techniques.

1.1 - Context and Motivation

According to the UNESCO Institute for Statistics definition [1], cultural heritage includes artifacts, monuments, a group of buildings and sites, museums that have a diversity of values including symbolic, historical, artistic, aesthetic, ethnological or anthropological, scientific, and social significance. It includes tangible heritage (TCH) (movable, immobile and underwater), intangible cultural heritage (ICH) embedded into cultural, and natural heritage artifacts, sites, or monuments. The definition excludes ICH related to other cultural domains such as festivals, celebrations etc. It covers industrial heritage and cave paintings.

Cultural heritage is a constantly evolving concept that bridges the gap between the past and the future through action in the present. It is rooted in sophisticated historical processes and serves as an irreplaceable testament to society's ever-changing value systems, symbolically representing the cultural and natural identities of many communities.

Expressing itself through monuments, artifacts, landscapes, and even intangible practices (like music and gastronomy), it greatly contributes to the formation of a community's identity and the unification of cities, countries, and the world, with categories like national and world heritage. The acknowledgement of this historical narrative modeling allows cultural heritage

objects to create a sense of integration from the traditional activities centered around them. Interestingly, the decision to preserve particular objects, monuments, or natural environments shapes the trajectory of these cultural narratives, significantly influencing a specific community's perspective about the past and the present.

While the term “cultural heritage” has been shaped by the values different societies have given to monuments, buildings, works of art, and so on, it also speaks to a broader issue: the systematic destruction or loss of irreplaceable objects. Phrases like “outstanding universal value” arose from the understanding that cultural heritage objects are unique and virtually irreplaceable [3].

In the Late Medieval and Early Modern periods, cultural heritage already started to take shape, through the collection of small objects such as artwork and other cultural masterpieces, named as “antiquarian interest”, marking the early stages of institutionalization. The basis for these selections was tied to the inherent value, rarity, and aesthetic quality of the object [2].

The concept of national heritage, which gained popularity in the 19th century, played a key role in propelling the preservation of cultural heritage, leading to the establishment of national museums and commissions for their protection. Furthermore, the emergence of organizations such as UNESCO shows that international consensus could bring about unified efforts to preserve important parts of a nation's and the world's cultural heritage.

The unfolding of a more holistic perspective and a strong, critical approach to understanding cultural heritage made significant progress in the late 20th century, primarily due to the development in practices [2][3][4]. But with progress comes adversity, as recent developments and specifically the abuse of the concept, have led to a strong critical approach in the context of heritage business, even playing a negative role in resurrected nationalist movements, extremist, and fanatic organizations. Deliberate destruction of heritage values and objects on one hand, and distorted, ahistoric or propagandistic interpretations on the other, can be found in various parts of the world, influenced by various ideologies, religious or political movements.

In the 21st century it is better appreciated that heritage embodies multiple meanings, benefiting from methodologies sourced in various disciplines that can be universally developed and implemented. Disciplines as diverse as humanities, social sciences, and environmental studies all take an interest in cultural heritage. The complexity of resource management in this sector emphasizes the need for recognizing and legitimizing differing interests to find a common ground. This wider approach can aid in navigating the often-challenging field of culture preservation.

As we progress further, a field that holds great potential for breakthroughs in this area is Artificial Intelligence (AI). It offers new methods that could complement current practices, opening up possibilities of enhanced recognition, conservation, and interpretation processes. AI can potentially become an instrumental tool in safeguarding our shared cultural inheritance, helping to celebrate the past and build the future.

1.2 - Objectives and Contributions

The objective of this master's thesis is to develop and evaluate an image analysis system based on AI, specifically conceived for preservation, restoration, and accessibility of cultural heritage assets. This system, named "DeepRevive", aims to use state-of-the-art AI algorithms to automatically generate documentation of these assets and analyze them in terms of deterioration to facilitate their restoration and improve accessibility for researchers, historians, and the general public. The DeepRevive system is intended to automate the process of generating documentation by extracting valuable information from the images, such as metadata, artifact characteristics and historical context, ensuring comprehensive and accurate records and thus offering better search capabilities and accessibility.

With this in mind, the scope of this dissertation is centered around designing a Convolutional Neural Network (CNN), which is a popular model in the field of computer vision, specifically designed for processing visual data. This paper delves into different elements of designing a CNN, such as data acquisition, pre-processing, and augmentation, as well as the intricacies of architectural decision-making including transfer learning and fine-tuning of an existing network, regularization techniques and so on. Additionally, concepts like class weights, batch sizes and loss functions, as well as overfitting and how to optimize the whole training process will be explained.

Building on this network, this thesis also explores the object detection technology, which gives it more meaning and applicability, addressing techniques such as image pyramid and sliding window.

By providing detailed insight into each step incorporated in designing a CNN model and object detector, this study aims to mitigate the persistent opacity surrounding AI applications in cultural heritage. Moreover, through strong methodological research and analysis, it seeks to contribute to laying the foundation for future research and applications of AI in this field.

Therefore, this document serves as a comprehensive guide for researchers and practitioners in the intersection of AI technology and cultural heritage documentation, accessibility, and preservation.

1.3 - Organization and Content

This first chapter serves as an introductory contextualization to the concept of cultural heritage and as motivation to what the field might stand to gain from the research and application of AI.

Following this, the second chapter gives a deeper understanding of the concepts of Artificial Intelligence, along its relevant subfields of Machine Learning, Deep Learning (DL), and Computer Vision, as well as a literature review and state of the art survey, establishing a foundation for where this research stands, and what techniques can be further explored.

The third and most central chapter outlines the adopted methodology, diving deeper into the design of a CNN and subsequent object detector, and how these can be applied to this specific field.

Afterwards, the fourth chapter lays out the final results, further explaining the whole iterative process of experimentation to achieve an ideal solution that adds great value to the field of cultural heritage, and presenting the outcomes of the designed CNN model and object detection techniques' application to the task at hand.

Finally, the Conclusion section reflects on this thesis' contribution to the existing knowledge and gives an outlook on future research directions in AI-powered cultural heritage accessibility and preservation by addressing its flaws.

2 State of the Art

Many interpretations of artificial intelligence (AI) have emerged over the past few decades. In 2007, John McCarthy, often referred to as one of the “founding fathers” of AI, defined it as “It is the science and engineering of creating intelligent machines, particularly intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.” [5].

This debate may have gained popularity with McCarthy, but its roots trace back to an earlier time with the ground-breaking work of Alan Turing, widely considered the “father of computer science”. In his 1950 paper, “Computing Machinery and Intelligence” [6], Turing posed the thought-provoking question: “Can machines think?”. This originated the “Turing Test”, intending to distinguish between computer-generated and human responses. Despite being thoroughly investigated over the years, this test remains a vital part of AI’s history and continues to hold philosophical importance due to its linguistic implications.

Fast forwarding to more recent times, Stuart Russel and Peter Norvig’s “Artificial Intelligence: A Modern Approach” became a leading resource in the study of AI. In it, they classify AI into four categories, highlighting distinctions based on rationality and the contrast between thinking and acting. Turing’s definition aligns with their perception of AI as systems that think and act like humans [7].

At its core, Artificial Intelligence brings together computer science and extensive datasets to address complex issues. Machine Learning and Deep Learning, subsets of AI, leverage algorithms to construct systems capable of making predictions or classifications based on input data.

Over the years, AI has undergone many hype cycles, and the recent release of OpenAI’s ChatGPT is no exception. This undoubtably marks a significant development in the realm of natural language models, resonating with the earlier breakthroughs. These generative models go beyond learning the intricacies of a language, also being able to comprehend the structural rules of software code, molecules, natural images, and many others.

The potential applications of AI are increasing by the day, and we are only scratching the surface. Yet, as its usage becomes more prominent, the importance of addressing ethical concerns increases.

2.1 - Machine Learning

Machine Learning (ML) is a branch of artificial intelligence that focuses on creating algorithms that progress naturally over time by benefiting from experience and usage of data. In essence, ML allows machines to learn from data and execute actions or make predictions without any explicitly programming for that function.

Essentially, ML develops and applies algorithms which assist in such actions or predictions. These algorithms are designed to enhance themselves progressively, refining their accuracy and effectiveness with the increasing flow of data [8].

Conventional programming models function by following a predefined set of instructions to carry out a task. Machine Learning introduces a new dynamic to this framework by presenting a set of examples (data) for the system to complete an assignment, leaving it to the machine's "intelligence" to figure out the process based on these examples.

Let's consider a scenario where we want a machine to identify images of cats. Instead of rigid instructions defining the appearance of a cat, we provide it with numerous images of cats and rely on the algorithm to distinguish common patterns and features that constitute a cat. This algorithm becomes increasingly better at recognizing cats, even when faced with never seen images, as it processes more data.

Being capable of learning from data and improving over time makes machine learning extraordinarily powerful and adaptive. It has been instrumental in various technological advancements we see nowadays, ranging from voice assistance to autonomous vehicles.

2.1.1 - Types of Machine Learning

The field of Machine Learning is broad and varied, encompassing many different techniques and approaches. Not all algorithms learn, behave, or predict in the same way [8][9]. Generally speaking, ML tasks and techniques can be grouped into three main categories - supervised learning, unsupervised learning and reinforcement learning.

- Supervised learning is a type of Machine Learning where the model is trained on labeled data. In other words, during training, the model is provided with inputs and the correct outputs. The goal of this model is to learn a mapping function from the inputs to the outputs. Once trained, it can be used to predict the output when given new unseen inputs. Some common types of problems in this type of ML include classification and

regression. In classification, the model learns to categorize data into predefined classes, as opposed to regression, where the model predicts a continuous output.

- Contrasting with supervised, unsupervised learning algorithms are trained on unlabeled data. Without the guidance of predefined outputs, these algorithms learn to identify structures and patterns that exist in the input data. This can involve tasks such as clustering, where the model identifies groups of similar data samples, or dimensionality reduction, where it simplifies the data without losing key information.
- Reinforcement learning operates differently. Here, the model learns how to behave in an environment by performing actions and observing results. It gets rewards for performing correctly and penalties for making mistakes. The goal of this type of learning is to achieve an optimal policy that maximizes the cumulative reward for the model over time. A common application is learning how to play a simple game, for example snake.

Deep learning, often considered the cutting-edge of Machine Learning, is a subset that focuses on the employment of sophisticated, multi-layered Artificial Neural Networks (ANN). ANNs are architectures inspired somewhat by the interconnected structure of neurons in a human brain.

DL algorithms deviate from traditional linear interpretation of Machine Learning, as they construct much more intricate, non-linear, and hierarchical models. They have the capacity to independently learn features from a vast amount of data by transforming inputs across multiple layers, where each layer learns to transform its input data into a slightly more abstract and composite representation.

One of the key advantages of Deep Learning is its ability to process and learn from large datasets. As the volume of data increases, its performance continues to scale, making it uniquely suited to exploit the wealth of information in the current digital age.

These algorithms have been successfully deployed in numerous applications, often significantly outperforming traditional Machine Learning approaches. For example, Deep Learning has been used to achieve state of the art results in tasks such as language translation, where a neural network learns to transform sentences from one language to another. In image recognition, Deep Learning can identify and categorize images with an accuracy that rivals, and sometimes exceeds, human ability.

In the field of personalized medicine [9], DL algorithms can analyze a patient's medical records to predict outcomes and suggest personalized treatment plans. These are just a few examples of the revolutionary applications and capabilities that have made deep learning an exciting field in the world of Artificial Intelligence.

2.2 - Computer Vision

In the process of developing Machine Learning systems, there are advantages to limiting its scope of operation. Rather than developing algorithms that learn and operate in multiple environments, data scientists tended to use ML for a particular task, as it could reach higher performances.

Yet, for many applications, the objective might be for Artificial Intelligence to be versatile and usable across various realms. This concept of a more general and wider AI has long been a standing vision of scientists since the dawn of computer science. This led to a surge in research during the 1970s and 1980s, aiming to infuse Artificial Intelligence capabilities into areas like image processing, language recognition, and robotics.

Implementing AI into disciplines such as robotics poses challenges, primarily due to the need to absorb, comprehend, and react to visual data. Hence, advancements in areas like sophisticated visual and auditory sensors, environment navigation systems, and improved mobility tried to keep up with the evolving times. And this demand for hardware that supports visual data collection has increased thanks to advancements in high-performing computing, big-data [8], and cloud systems.

This led to the genesis of Computer Vision - a specific Deep Learning application that allows computers to understand visual input with a high degree of accuracy, using information from digital images and videos to execute informed decisions.

Like most machine learning components, CV also requires extensive datasets to enable the algorithms to decipher this information correctly. It primarily uses two techniques:

- Deep Learning, as mentioned earlier, facilitates solving complex problems. Coupled with Neural Networks, DL can essentially teach a machine's "brain" to process visual data, recognize patterns, and adapt to environment changes over time.
- Convolutional Neural Networks, on the other hand, leverage convolutions (a process that forms a mathematical function from two other functions) to make predictions about images.

In short, Computer Vision utilizes CNNs and DL to conduct rapid and high-volume learning on visual data. This training enables machines to interpret data, similarly to how a human eye functions, further enhancing these systems' intelligence [10][11].

Following this, we will dive deeper into the workings of these Artificial Neural Networks, and how Deep Learning harnesses their complexity and computing power to transform industries and make breakthroughs across multiple domains.

2.3 - AI for Cultural Heritage

As we delve deeper into the digital age, Artificial Intelligence is increasingly influencing various aspects of human life, including the preservation and restoration of our rich cultural heritage. More specifically, the subfield of AI known as Computer Vision offers valuable tools to help us accurately archive, restore, and preserve historical landmarks from the past and present.

A powerful example of this was seen in the aftermath of the devastating fire at the Notre Dame Cathedral. CV technology was used to help reconstruct the iconic structure, using a precise 3D representation of the cathedral's architecture made before the event, using laser digitization to capture over a billion data points. This digital representation of cultural monuments, coupled with AI design tools, is revolutionizing the work architects do. Computer Vision assists not only in modern design planning, but also paves the way for immersive virtual reality experiences, providing an engaging method to explore reconstructed historical sites.

However, the integration of AI into cultural heritage management comes with its obstacles. Many state-funded institutions face challenges, such as potential copyright infringement issues and inefficient digitization processes. This is where the power of CV can make an impact. By optimizing the archival task and transforming user engagement, it can give new life to heritage preservation.

Moreover, the introduction of funding from the European Union, together with advancements in generative AI, are promoting technological interventions in cultural preservation. Initiatives like Europeana [12] are pushing for a wider digitization process and improved accessibility to European culture, promoting concepts like the creation of a shared European data space.

Computer Vision also plays a crucial role in addressing copyright issues, especially with the ongoing growth of datasets which are essential for conservation projects. Automated recognition and attribution of copyrighted content becomes possible, assisting organizations in rightfully crediting the original creators.

Simultaneously, as this technology continues to evolve, new concerns, such as the production of synthetic media or "deepfakes" generated by AI, will rise. Thus, it is a process that demands caution, and the establishment of regulatory frameworks and ethical boundaries to abide by.

Also, it is worth noting that these capabilities and applications of AI, specifically CV, are not merely theoretical. Around the globe, numerous exciting projects are already in progress, leveraging these technologies to preserve and promote cultural heritage, addressed in the following sections.

To sum it up, yesterday's treasures should meet today's advanced technology, implementing these contemporary tools and techniques dynamically, respecting both tradition and innovation.

2.3.1 - CNN

As previously stated, Machine Learning systems fundamentally differ from “traditional programming”, in that they “learn” from input data, rather than following a pre-determined set of instructions. They do so through the implementation of a Neural Network, which is made up of interconnected artificial “neurons”. We can think of each neuron as something that holds a number, say 0 or 1, saying if it is active or not, much like our human neurons fire electrical signals between each other.

Each neuron comes with its weights and a bias, which are key to the learning process. It is these values that are updated during the long trainings of the network. Weights represent how influential a given neuron is, while biases allow for threshold establishment. This all comes together in the neuron, that takes in the input signal, multiplies it by its weight and adds its bias, before passing this new value through a so-called activation function, that serves the purpose of limiting the intervals of values outputted by the neuron. Commonly, this interval is $[0, 1]$, which can be accomplished by the sigmoid or softmax functions. And this is just one neuron.

As illustrated in Figure 1, all the neurons are organized into input, hidden and output layers, and in their most basic form, constitute a multilayer perceptron. The input layer receives the raw input signals. Generally speaking, it has as many neurons as there are input features, and for image analysis, many models consider the number of input neurons to be equal to the number of pixels in that image.

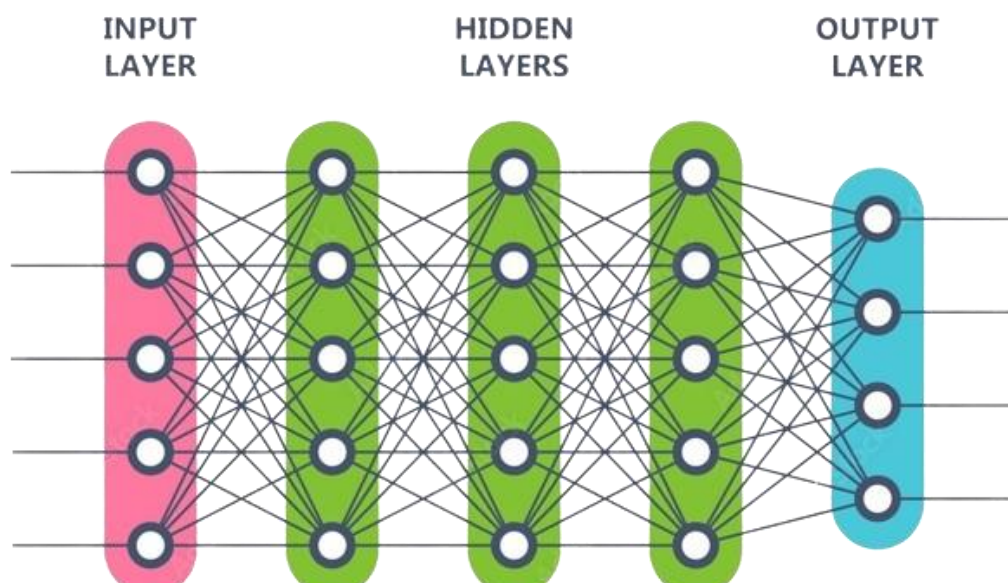


Figure 1: Illustration of a simple multilayer perceptron (Source: Web)

The hidden layers are responsible for the majority of the computations. There is a lot of space to experiment with the number of neurons on these layers, as they do not have any limitation. And finally, the output layer delivers the results. Again, they have as many neurons as we want there to be outputs. In the example of image analysis, if we have 10 classes to choose from, there will be 10 output neurons. But how do they learn?

Using supervised learning as an example, after the network's structure is complete, it is submitted to a training process on a large, labeled dataset. It randomly initializes all the weights and biases and makes a prediction on a given point of data, comparing it to its respective label. To quantify this difference, we have to define a loss function, that can vary from case to case. But this tells the network how far or how close it was to the expected outcome, and the process of training the network simply tries to minimize this loss function, employing an optimization technique called gradient descent. The loss function's derivative concerning each weight and bias is calculated, and then the weights and biases are slightly tweaked in the opposite direction of the gradient, which is the fastest way to reduce the loss function. Repeating this with numerous input samples slowly improves the model's accuracy. This whole process is known as Backpropagation [13].

When we reach Convolutional Neural Networks, they revolutionize this basic process of Neural Networks by introducing a spatial/sequence awareness aspect that fully connected layers do not have. It is one of the state-of-the-art techniques being used in the field of Computer Vision. These networks are designed to both automatically and adaptively learn the images' features to ultimately recognize patterns with a high degree of abstraction. The typical CNN basic structure is made up of three types of layers: convolutional, pooling and fully-connected, as seen in Figure 2.

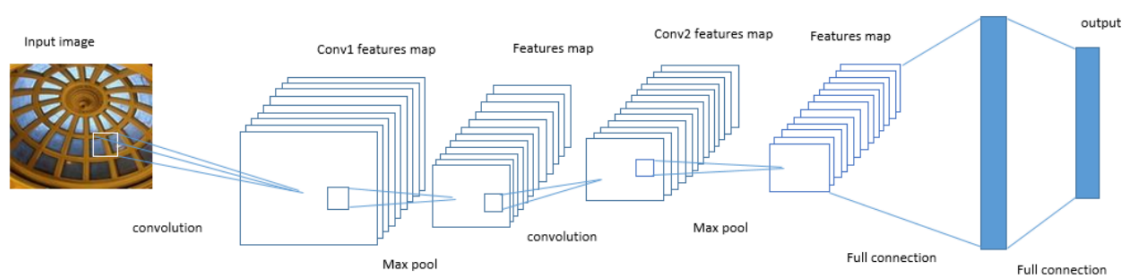


Figure 2: Illustration of a simple CNN (Source: [14])

The neurons in the first layer are not connected to every single pixel. Instead, it works by sliding a filter (kernels) across the input image and execute several matrix multiplications that result in the feature map (known as convolution). This filter has some associated weights, which are the main parameters that are adjusted during the training phase.

The pooling layer, or subsampling layer, follows the convolutional layer. It reduces the dimension of each feature map, similarly to the previous layer, by sweeping a filter across the input. It differs from the convolutional layer because the filter does not have any weights. Instead, it takes the maximum (Max pooling) or the average (Average pooling) pixel values, hence reducing the dimension of the maps.

Lastly, the fully-connected layer receives the flattened output from the previous layers and computes the final classification. While the first two layers usually use ReLu as their activation function, fully-connected layers use a soft-max to produce a probability from 0 to 1.

Therefore, CNN models, just like other neural networks, deduce ideal values for weights and biases, but they do so in a way that takes into account the spatial hierarchy of the input data. Understanding how these components fit together is key to grasping the power and flexibility of Convolutional Neural Networks. By combining linear algebra operations with non-linear activation functions, CNNs can learn high-level features from raw pixel data, being able to handle vast amounts of data, recognize complex patterns and generate valuable predictions that traditional programming approaches cannot. This is why CNNs are a driving force in computer vision tasks [14].

2.3.2 - Other techniques

Other than CNNs, another popular technique is Recurrent Neural Networks (RNNs), that have also found their way into the image analysis world. RNNs are predominantly used for sequence prediction problems, considering they can use their internal state (memory) to process sequences of inputs, which gives them advanced capabilities for tasks like natural language processing, translation, speech recognition, and more.

Aside from the more traditional methods like CNN and RNN, there have been new methodologies introduced in recent years that look promising for the application in cultural heritage preservation, such as Generative Adversarial Networks (GANs). GANs can generate new instances of images that learn to mimic a given data distribution, which can be very useful in creating reconstructions of damaged or partially lost cultural artifacts.

Moreover, more complex techniques like the Mask R-CNN have been successfully used for instance segmentation to not just categorize objects withing an image, but also precisely identify boundaries of each individual object, which has promising applications for the detailed analysis of intricate artworks, archeological sites and architectural structures [10][11].

While these models and techniques have significantly advanced our ability to analyze and classify images, it is important to remember that each comes with its own set of strengths and limitations. Hence, the choice of model should always be dictated by the specific requirements and constraints of the task at hand.

Advancements in Deep Learning and AI are continuously propelling the development of newer and more efficient models and techniques. Therefore, the existing models should be viewed as steppingstones towards the future landscape of AI and cultural heritage preservation. This relentless innovation is sharpening the discipline, offering enhanced precision, superior objectivity, and a far broader spectrum of applications.

The game changer in this field, however, is the integration of these AI models with traditional techniques of cultural heritage preservation, a fusion of technology and tradition. How these models can be effectively applied to transform the state of cultural heritage, and to what extent, remains a question that researchers and professionals are currently exploring.

2.3.3 - Existing projects and Research Opportunities

The field of AI for cultural heritage preservation has some notable projects which make use of popular, powerful Deep Learning models, like the GoogLeNet and ResNet, both of which are derived directly from the CNN architecture. These models are heavily used in Computer Vision tasks due to their capacity to efficiently learn and recognize patterns in images.

Among these, GoogLeNet, also known as InceptionV1, marked a significant advance in the field by introducing the inception module, a complex building block with parallel branches of varying depth, capable of capturing both high-level and low-level features in images. In essence, its arrangement of convolutions and pooling, allows it to simultaneously learn different types of features, therefore providing a more comprehensive understanding of the content within the image.

ResNet, on the other hand, is more focused on addressing the problem of training very deep networks. It achieves this through the innovative use of skip connections, also known as shortcut connections. These connections allow the output of one layer in the network to be fed directly into a more advanced layer, bypassing the layers in between. The result is a network that can learn residual mappings instead of attempting to learn the full output mapping, significantly easing the training process, and also improving the network's performance. Due to its simplicity and effectiveness, the ResNet network has been widely adopted in the field of CV, and it often serves as the backbone in various complex systems. Having demonstrated good performance in computer vision tasks such as image classification, object detection and segmentation, the ResNet was the chosen network to build this study upon.

Within the scope of cultural heritage preservation, these models are increasingly being implemented. There are already some projects that successfully utilize these architectures for visual recognition tasks by analyzing historical images, artwork and more, demonstrating superior results compared to conventional Machine Learning methods. This success reiterates the potential of these Deep Learning models in aiding conservation efforts [16][17].

Taking a creative stance within the field, the MonuMai project [15] demonstrates the application of AI technology for the recognition and categorization of architectural monuments. This project uniquely integrates transfer learning and fine-tuning techniques on Resnet, coming up with their own network MonuNet. The implementation of these techniques enables the system to recognize architectural styles effectively, thus aiding in the management and preservation of important architectural monuments.

These emerging projects underscore the potential that combining technology with cultural heritage holds. Although still in the early stages, this fusion looks very promising. Particularly with the rapid advancements seen in the field of AI, and more specifically Deep Learning, there is the promise of even more powerful and efficient models. Such technological advancements are set to elicit significant transformations in the preservation and understanding of our cultural heritage.

Looking forward, the continued experimentation and application of models like this in the realm of cultural heritage preservation hold great promise, although there still exist several challenges and research opportunities.

Deep-learning models typically require large amounts of annotated data for training. Unannotated images of art and historical sites are abundant, but this lack of annotated data can and does limit the performance of such models. There is a significant need for efforts to construct large-scale, richly annotated datasets in general, and specifically for cultural heritage preservation applications.

Consider the project highlighted in the paper “Deep learning-based weathering type recognition in historical stone monuments” [16]. This study draws on hand-labeled data for a CNN model. The pictures and respective labels were all created by the researchers, which was not feasible for the current application. Nevertheless, it showcases how teams around the world are innovating solutions within the present limitations.

Furthermore, an interesting approach of structural mapping was used in “A semantic modeling approach for the automated detection and interpretation of structural damage” [17]. This study uses 3D and 2D point cloud data to create detailed digital maps of architectural sites for assessing structural damage. This involved using a drone to capture aerial photographs of the site, which were then turned into a 3D digital model via a technique known as Structure from Motion. This model then served as training for a CNN to identify and categorize different types of structural damage.

Despite the potential of these methods, their application in this work was not possible, as 3D model creation requires resources and access to sites. In addition, computational resources and specialized expertise are necessary to transform the images obtained into the required models, and this approach may not scale well to other forms of culture, such as manuscripts, paintings, or artifacts.

Another notable venture is Google Arts & Culture, a prominent platform that already uses machine learning technology to bring art and culture from over 2000 organizations worldwide to the fingertips of the ordinary online user. By digitizing and making art collections accessible, it showcases yet another significant overlap between tech and heritage and the potential lying inside.

Research in making the models more interpretable can be an essential future step. Even though the predictive performance of models like ResNet is impressive, their inner nooks and crannies are often not easily interpretable. Given the critical importance of understanding why a particular prediction or classification is made in this context, this poses a considerable challenge.

Most of the current models operate offline and process batch data. But some applications could benefit from real-time analysis, such as in the case of monitoring and controlling the environmental conditions affecting artifacts in a museum. Thus, developing models that can operate and make predictions in real-time could be an important achievement.

While current efforts focus predominantly on tangible heritage like artifacts and monuments, much of the world's culture is, nowadays, intangible, which encompasses traditions, language, music, dance, among others. There is room for innovation and research in developing machine learning applications to process and preserve this intangible cultural heritage.

Even small changes in the input can sometimes lead to large and unexpected changes in the output of Deep Learning models. Testing and improving the robustness of such models, particularly in varying and harsh environmental conditions in which many cultural heritage sites exist, is still an open research area.

Incorporating ethics early into AI applications for cultural heritage is vital. While AI models may help predict the decay of a historical monument, decisions on which sites to preserve could be influenced unfairly by the model's biases. More work needs to be done on understanding and mitigating AI's potential biases.

2.4 - Conclusion

In conclusion, while the process of marrying AI with cultural heritage preservation is relatively new, it holds great prospects. The blend of cutting-edge technology with age-old legacies opens doors to the way we interpret, value, and conserve our rich heritage.

Next, we will dive deeper into how one could implement these models tailored to the application at hand, how different training techniques might drastically change the desired outcome and try to build on some of the underdeveloped fronts.

The following chapter will try to elaborate the existing architecture of the ResNet model, by customizing it to document, identify and improve access to cultural artifacts, which proved to be a great research direction, as well as using this custom network to further explore the concept of object and style detection.

Thus, the main goal of this research is to identify the possibilities within the realm of image analysis, building on some of the previously mentioned studies by addressing their current flaws, such as the lack of annotated data, and layout a comprehensive strategy for correctly arriving at an optimal network.

3 Methodology

3.1 - Designing a convolutional neural network

In recent years, the synergy between Artificial Intelligence and heritage has unraveled incredible opportunities for understanding and preserving our cherished past, a core aspect of our shared global identity. While AI-driven techniques are taking the center stage in the preservation and analysis of cultural heritage, there are some limitations and knowledge gaps.

The pressing challenges include dealing with the scarcity of annotated data and cultivating models that are not only accurate but also interpretable and robust, while retaining real-time applicability. Despite these challenges, the use of AI to safeguard intangible heritage has been explored considerably less than preserving physical, tangible heritage, making for another research gap.

This chapter confronts said challenges head on by leveraging advanced deep learning techniques, specifically the ResNet model. The rationale behind the use of this network rests on its exceptional performance for object recognition tasks. The model can identify different objects within images accurately, which is pivotal to this dissertation.

Nonetheless, the ResNet model alone was insufficient to meet the project's goals. Enhancements with targeted techniques were crucial to help achieve them. A central part of the methodology was transfer learning, which tackled the shortage of annotated data in the field of cultural heritage. By learning on a pre-trained ResNet model educated on ImageNet (a dataset with more than 14 million labeled images), it capitalizes on the learned representations which save computational resources and significantly improve accuracy, when compared to training from scratch.

Furthermore, to ensure this model will effectively classify correctly having been trained on a sparse dataset, fine-tuning was adopted by unfreezing and retraining a few top layers of the ResNet model. This was complemented by regularization techniques like dropout layers and the use of class weights, to tackle the disparity of the number of images between classes, just

to name a few, which strengthened the model's generalization ability and helped avoid overfitting and improve its robustness.

Also, to improve the physical interpretability of the outputs, an object detector was implemented, using techniques like a sliding window and an image pyramid, bringing everything together and grating more insights into the developed work.

In light of the challenges and current gaps in the application of AI for cultural heritage preservation, the techniques included in this hybrid methodology were carefully chosen to improve the final model's performance, getting it a step closer to better preserving our tangible and potentially intangible cultural heritage.

In essence, this chapter contains an outline of the main steps that need to be taken towards the integration of the ResNet model to a custom application, as well as the development of an object detector, and how each parameter can greatly influence the outcome.

3.1.1 - Data acquisition

For the successful execution of this thesis, two different yet complementary datasets were sourced from the web, due to their appropriateness and relevance for the research goals.

The primary dataset for the classification task was obtained from Kaggle, courtesy of Ikobzev [17]. This dataset is an invaluable resource that contains labeled images of different architectural elements. The images were previously pre-processed and uniformly resized to 64x64 pixels, as seen in Figure 3, which is highly suitable for this application. This dataset served as the basis for the training of the ResNet model to ultimately perform object recognition.



Figure 3: Sample image of the classification dataset (bell tower)

For the style identification, the dataset was sourced from the MonuMai project repository on GitHub [18]. The motivation behind the choice of this dataset was its rich collection of images, exemplified in Figure 4, representing various styles of historical monuments and

artifacts, which allows for exploring a wide range of cultural heritage expressions across space and time.



Figure 4: Sample image of the style identification dataset (gothic)

The selection of both datasets was carefully made to ensure a diverse and representative collection of images consistent with the architectural and style elements this model seeks to classify, even though the number of images of each class was not particularly consistent, but this is something that was overcome with the use of class weights.

Beyond sourcing, the next step involved additional data processing to make the images suitable for feeding into the ResNet model.

Table 1: Element classification dataset

Class	No. of images
Altar	828
Apse	505
Bell Tower	1057
Column	1913
Dome (inner)	589
Dome (outer)	1175
Flying Buttress	405
Gargoyle	1562
Stained Glass	998
Vault	1097

Table 2: Style identification dataset

Class	No. of images
Baroque	516
Gothic	360
Hispanic-Muslim	328
Renaissance	313

3.1.2 - Data pre-processing

Before feeding the images into the ResNet model, several pre-processing steps were required to ensure the images were in a suitable format to optimize the performance of the model.

The first pre-processing step involved setting the input shape for the model to accept and process the image data, which is crucial. Given that the dataset had already been processed to a uniform size of 64x64 pixels, the ResNet model was configured to accept this size, by specifying the input shape of the first layer as 64x64 pixels for all images entering the model. This standardization of input shape facilitated the smooth feeding of images into the network without any dimension mismatch.

Next on the list is normalizing the pixel intensity values of the images. Pixel intensities typically range from 0 (black) to 255 (white). For efficient training of the network, these intensity values were standardized so that they would fall within the range of 0 to 1. Normalizing pixel intensities is a standard technique in image analysis, particularly helpful in differentiating details in the images during the training phase. This was achieved by dividing the pixel intensities by 255, aiming to lessen computational requirements and to ensure stability during the training of the convolutional neural network.

In certain cases, overfitting could happen if the model sees the same images too many times during training. To mitigate this, data augmentation methods were implemented to add diversity to the data without needing new images.

Having been resized and normalized, the images were now in the ideal format to be adequately processed by the model. These measures allowed it to analyze the images more effectively and reduce the chances of hindrances during the training phase, therefore aiding in more accurate object recognition of different architectural heritage elements and styles.

3.1.3 - Data augmentation

Given the diversity inherent to cultural elements and styles, it was necessary for the model to be able to recognize a vast range of patterns. However, it is often challenging to collect a sufficiently large and diverse dataset to cover all possible variations. To overcome this, data augmentation techniques were employed. These techniques can generate new altered versions of the images, adding more diversity and quantity to the dataset, thus enhancing the generalization ability of the model.

This can be achieved by putting together several layers using a sequential model of python's Keras library, which then are added on top of the ResNet model. Some examples of possible augmentation layers are:

- Random flip: This operation performs a random horizontal flip of the image. This means that the image is mirrored along its vertical axis, effectively providing a view from a

different perspective. This is particularly useful in countering any positional bias and helps the model to recognize objects irrespectively of their orientation.

- Random rotation: Rotating the image by a random factor within a range. In this case, the rotation factor was set to 0.1, which corresponds to 36 degrees. Random rotation creates variation in the image orientation, increasing the model's robustness to the position and angle of objects in the image.
- Random zoom: Randomly zooming into the image by a certain amount, in this case by 10%. Zooming alters the scale of the elements in the image, making the model more resilient to size variations in the real-world representation of these elements.



Figure 5: Example of images generated through data augmentation

As seen in Figure 5, the aforementioned layers create variations in position, angle, and size, making the model more robust and general. However, there are several other types of data augmentation, including:

- Brightness augmentation: Altering the brightness of the image can make the model robust to different lighting conditions.
- Shear transformation: This pushes pixels along an axis creating angled distortions, which can be useful to generate geometrically different views of images.
- Random cropping: Randomly cropping images can expose the model to different parts of an object.
- Noise injection: Adding random noise to pictures can help the models be more resilient to visual fidelity or artifacts from image sensors.

In summary, data augmentation methods provide a way to artificially expand and diversify the dataset, creating a better opportunity for the model to learn broader features and, therefore, improving the overall performance of the whole network.

3.1.4 - Transfer learning and fine-tuning the ResNet network

To tackle the challenge of accurately detecting and classifying architectural elements from the image dataset, transfer learning with the ResNet network was used. This choice was predominantly motivated by ResNet's powerful ability to discern patterns in image data and its unprecedented success in tasks related to image recognition.

ResNet, short for Residual Networks, is a type of Convolutional Network architecture that sparked notable advancements in the field of Deep Learning. Particularly, this model is distinguished by its deep layer architecture that successfully resolves the vanishing gradient problem commonly encountered in training deep neural networks. This unique architecture includes "skip connections" or "shortcut connections", which allow the model to skip certain layers, making it easier for the network to learn an identity function and facilitating the training process.

The version of the ResNet model that was used was pretrained on ImageNet, a vast image database designed for use in visual recognition software research. This pretrained model has learned robust features for identifying a thousand different classes of images by being trained on more than 14 million images.

The concept of transfer learning forms the central point of the adopted methodology. It refers to transferring learned features from one model to another. This was specifically employed to capitalize on the feature extraction capabilities of the pretrained ResNet model. This step was particularly important because the pretrained model had already identified universal features across its large training dataset, which could be very useful for a smaller target dataset.

After importing the pretrained ResNet model, a process called fine-tuning was carried out to adapt the model to the specific task at hand. Here the lower layers responsible for extracting universal features were frozen, meaning prevented from retraining, while top layers were unfrozen and retrained on the acquired architectural elements dataset. The reasoning for this decision is that low-level features, such as edges and textures, are much more common across a broader range of images, while higher-level features are more application specific. Therefore, the top layers were fine-tuned to learn features from this niche field: our cultural heritage.

Adding to this, a few more layers were added on top of the network model (these included a global average pooling layer to mop up spatial information and a densely connected layer to output probabilities for our specific classes) to ensure that the output of the model would be suited to this classification and detection task.

In conclusion, taking advantage of a pretrained ResNet model combined with the fine-tuning of the model's top layers and incorporating several more layers at the top, allows for the possibility of creating a powerful model that could effectively handle the intricacies of this project's goal.

3.1.5 - Regularization

To ensure the optimal performance of the ResNet model and prevent overfitting, some regularization techniques were adopted. Overfitting is a common problem in machine learning where a model learns the training data too well, capturing the noise along with the underlying feature patterns. While this leads to an excellent performance on the training data, it performs very poorly on unseen data. Regularization provides a means to tackle this issue, ensuring that the model generalizes well on new images.

One of the key regularization techniques used is dropout. Dropout is a simple yet effective method of preventing overfitting in neural networks. During the training process, some neurons tend to overpower others and “dominate” the learning. The insertion of dropout layers combats this issue by randomly “dropping out” or turning off a fraction of the neurons during each training epoch, which forces the model to distribute the learned information across all nodes.

In this implementation, dropout layers were added after some layers in the model to randomly drop a certain percentage of neurons, which was experimentally determined. This essentially created a form of ensemble learning, where each training instance is exposed to a unique set of neurons. At the testing stage, all neurons were used but their outputs are scaled down by their dropout rate, to match their behavior during training.

Although this is a fine way of significantly reducing overfitting, using it excessively could lead to underfitting. Underfitting occurs when a model is too simple to capture the underlying structure of the data. If too many dropout layers are incorporated into the network, or the dropout rate is set too high, it can lead to a scenario where too many neurons are turned off during training. This can immensely affect the model’s ability to learn from the dataset. Consequently, its complexity is reduced, causing it to perform poorly not just on the training data but also on the validation data.

Besides dropout, another technique of regularization was used, called early stopping. This implies the monitorization of the model’s performance on resisting the validation set. As soon as the validation loss stops improving for a certain number of epochs, training is halted, with the use of a callback function. This also prevents the model from overfitting the training data and aids in generalizing better to never seen data.

After setting up dropout layers and early stopping as the regularization techniques, the setting of the number of epochs was in order. An epoch refers to one complete pass of the entire training dataset through the whole network. When adopting early stopping, it is common to set a large number of epochs and let the model decide when to stop the learning to get it optimally trained. This approach allowed the learning process to run for as long as there were improvements being made in the model’s performance on the validation set, which is a common practice.

In summary, the blend of dropout, early stopping and a high number of epochs made the regularization strategy complete. Together they helped the model to prevent overfitting and underfitting, allowing it to learn just enough to generalize well to unseen data without memorizing the noise and details in the training data, a so-called appropriate fitting.

3.1.6 - Batch size

In machine learning, particularly in deep learning, batch size is one of the vital hyperparameters that dictate how the training process is executed. Batch size refers to the number of training samples processed before the model's internal parameters are updated.

The choice of batch size can significantly affect the model's performance, the memory requirement, and the total time used for training. If the batch size is too small, the model will take longer to converge and might get stuck in local minima, leading to a sub-optimal solution. On the other hand, too large of a batch size may result in rapid convergence, but also to a sub-optimal solution. Additionally, larger batch sizes require more memory to store the information of the entire batch.

Keeping this in mind, the choice for the specific batch size in this model was based on a balance between computational efficiency (both in terms of speed and memory usage) and its performance. The goal was to set a batch size that was large enough to enable efficient hardware usage (the training process was run using a GPU, which performs better with larger batches) and to provide a reasonable approximation of the gradient descent, but also small enough to avoid excessive memory use and to ensure that each step involves a degree of noise so as to avoid getting stuck in sharp, non-optimal local minima.

Various values of batch sizes were experimented with to get to an optimal balance. During these experiments, the model was checked to assess how well it was learning from the data and how quickly it was converging. Typical values for batch sizes are powers of 2 ranging from 16 to 1024.

In essence, the specific batch size used was a trade-off between memory limitation, computational speed, and the network's ability to learn and generalize.

3.1.7 - Added layers

As part of the model optimization process, some extra layers were added to the pre-trained ResNet network, to specifically tailor it to the dataset and task at hand. The ResNet model was fantastic at extracting meaningful features from the images, but some further customization of the final layers was needed to generate the output suitable for this classification task.

The primary reason for adding layers to the ResNet was to adapt the model's output dimension to this project's needs. This network was initially trained to classify images into

1000 classes, but this specific use-case only has 10. Thus, the model was loaded without its final classification layers and some customized layers were appended to perform this classification task. As such, the final layers of the original network, which were specific to the ImageNet dataset, were removed, and new ones were appended.

Here is a brief overview of the added layers:

- Global average pooling layer: After the convolutional layers of ResNet, a global average pooling layer (GAP) was added. The GAP layer reduces each h, w, d feature map to a 1D vector by simply taking the average of all height (h) and width (w) value pairs for each depth slice (d). This has two considerable advantages: it helps reduce overfitting by minimizing the total number of parameters in the model, and it makes it more robust to spatial translation of the input images, which is very useful in combination with the use of data augmentation techniques.
- Dense layer: After the GAP layer, a dense (or fully connected) layer was added. This layer is needed as it connects all the neurons in the previous layer to every single neuron of the next, effectively learning global patterns in its input feature space.
- Dropout layer: To further avoid overfitting, a dropout layer was added after the dense layer. As explained before, this randomly switches off some neurons in the layer, which forces the weight to distribute more evenly, leading to a less weight reliant and more general model.
- Output layer: The final layer is also a dense layer, with a number of neurons equal to the number of classes. Its role is to output the probability distribution over all the classes. The softmax activation function was used in this layer, which is standard for multiclass classification problems. This function takes as input the values of each neuron and normalizes it to be in the range of 0 to 1 (probability), illustrated in Figure 6. The neuron with the highest probability defines the output class.

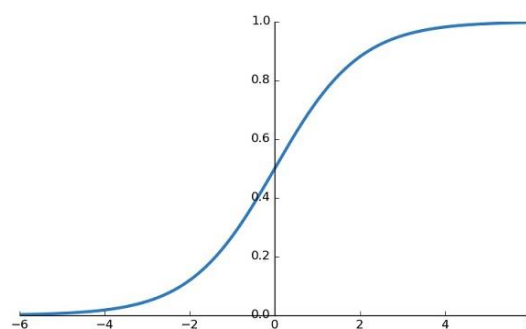


Figure 6: Softmax activation function

Several pairs of dense and dropout layers were implemented before the output layer. Adding more dense layers with the ReLu activation function increases the depth of the model, leading to a more refined level of abstraction.

The Rectified Linear Unit (ReLU) activation function is commonly used in neural networks and deep learning models. This function returns 0 if the input is negative, and the input itself if it is positive, as seen in Figure 7. It has several advantages:

- Non-linearity: Although it looks like a linear function, the ReLU has a derivative everywhere except zero, allowing it to act as a non-linear function. This non-linearity helps deep learning models learn from error.
- Computational simplicity: The ReLU is simple and computationally efficient compared to other activation functions (like sigmoid or tanh) because it can directly pass positive inputs and set negative ones to zero.
- Addressing the vanishing gradient problem: Deep learning models often suffer from vanishing gradients, where the gradient of the loss function approaches zero as the model trains. This makes the network hard to train, as the weights and biases of the initial layers are barely updated. ReLU mitigates this problem, because its gradient is either 0 (for negative inputs) or 1 (for positive inputs).

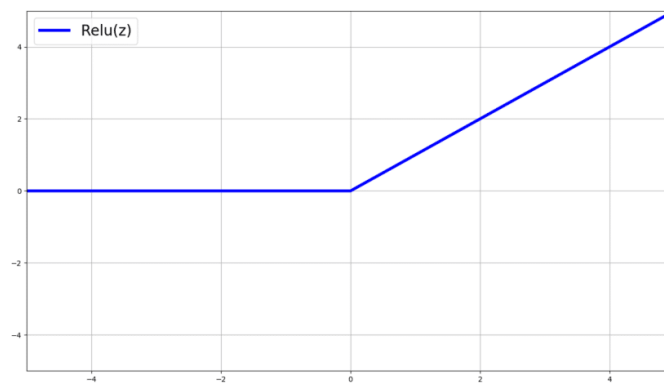


Figure 7: ReLU activation function

However, it is also worth pointing out that the ReLU can have an issue called “Dying ReLU”, where some neurons can stop learning due to consistently receiving negative inputs and therefore always outputting 0. This can be mitigated by using variants of ReLU, such as the Leaky ReLU or Parametric ReLU.

It is a common practice to gradually decrease the number of neurons in the network’s successive layers, starting with a larger number and narrowing down towards the number of classes. This architecture reflects the belief that many low-level features (captured in the first few layers with more neurons) combine into fewer high-level features in later layers. Here, the dense layers in the hidden layers were designed to gradually decrease from 512 to 256, then 128 and finally 64.

Moreover, reducing the number of neurons as the model develops towards the final layers, helps increase computational efficiency, decrease the risk of overfitting by reducing the total number of adjustable parameters, and helps keep the focus on the most critical features.

These added layers allow the model to learn more complex representations by creating non-linear transformations of the input. Each dense layer with ReLu transforms the learned representations from the previous layer and passes them to the next one.

As the model becomes deeper, the likelihood of overfitting increases, because the model might start to learn the noise in the training data. To counteract this, once again, dropout layers are used after each dense layer.

This architecture choice of having alternated dense and dropout layers is a common practice to both increase the model's capacity (by adding depth) and prevent overfitting (by adding dropout).

In essence, these added layers allow the preservation of the powerful feature extraction capabilities of the earlier layers in the ResNet network, while effectively tailoring the architecture to this specific task. These modifications were vital in ensuring a good performance on the dataset.

3.1.8 - Optimizer and learning rate

In the process of training the neural network, the optimizer and learning rate play critical roles.

The optimizer's function is to adjust the attributes of the neural network, such as weights and learning rate, to minimize error and loss. The selected optimizer was Adam (adaptive moment estimation) because it combines two other effective optimization techniques: RMSprop (root mean square propagation) and stochastic gradient descent with momentum.

Adam calculates an exponential moving average of the gradient and the squared gradient, having the decay rates of these moving averages controlled by two parameters β_1 and β_2 . Given the benefits of AdaGrad's handling of sparse gradients, and the advantage of RMSProp's handling of non-stationary objectives, Adam tends to work well on a wider range of problems, compared to other optimizers.

The learning rate determines how big the steps the optimizer takes to find the minimum loss are. It is one of the most challenging hyperparameters to define, because it significantly influences the model's performance. A higher learning rate allows the model to learn faster, at the cost of arriving to a sub-optimal set of weights and biases. A smaller one may allow the model to learn a more globally optimal solution but may take significantly longer to train and may also get stuck in a plateau region.

The optimal learning rate highly depends on the nature of the problem and the model's architecture. Different values were experimented with before picking the initial value of 0.00013 that allowed the training to converge well.

It is also worth noting that the Adam optimizer has a self-tuning mechanism for the learning rate. It adjusts the learning rate dynamically based on the gradient's moving averages, leaving

the developer with less of a burden to manually tune this hyperparameter. This feature is one of the reasons why Adam is such a popular choice among deep learning tasks.

Through these strategic choices in optimization and learning rate, the model was able to learn from the data accurately, efficiently and in a reasonable amount of training time.

3.1.9 - Loss function

In supervised learning, the loss function (also known as the cost function) essentially measures the inaccuracy of the predictions made by the model during the training phase. Its choice is critical and depends largely on the problem at hand. For instance, if one is solving a binary classification problem, the binary cross-entropy loss function is a common choice. For multi-label classification problems, the categorical cross-entropy loss function is typically used. The mean squared error or mean absolute error functions are commonly used for regression problems, where the task is to predict a continuous outcome.

In this case, the problem is of a multi-class classification (as indicated by the softmax activation function in the final layer), where the categorical cross-entropy loss function is typically used. This function calculates the individual loss for each class label per epoch and sums them. In other words, it calculates the difference between the predicted probabilities and the expected outcome.

While the main goal during the model's training phase is to minimize this loss, the optimization of the loss function does not guarantee an increase in accuracy. The loss measures how far off the entire set of predictions are, whereas accuracy reflects the proportion of correct predictions. As such, reducing the loss corresponds, in most cases, to an increase in accuracy, because smaller differences between predicted and true values imply more correct predictions.

However, the relationship between loss reduction and accuracy increase is not always linear. Given that accuracy is a purely binary measure, while loss assesses how far off individual predictions are (which can be fractional), a model could be "almost right" on many instances, leading to a low loss, but if it still gets those instances wrong, accuracy remains unchanged.

The choice of loss function, as well as the optimizer and learning rate, collectively impact on how quickly and well the model can learn from its mistakes. The proper consideration and selection of these, taking into account the specific requirements and constraints of this use case, is one of the most central aspects of successfully tuning the model's hyperparameters to ensure success.

Nonetheless, it is essential to remember that model performance encompasses more than just minimizing the chosen loss function. Both loss and accuracy offer illuminating, yet different, perspectives on how well the model is performing. Monitoring both these metrics during model training can contribute to a more comprehensive performance evaluation. It can

also provide assistance in identifying potential issues such as underfitting or overfitting, thereby providing valuable feedback for subsequent hyperparameter tuning.

3.1.10 - Class weights

In machine learning tasks, particularly in classification problems, it is not uncommon to encounter datasets with imbalanced classes, where some classes have significantly more samples than others. This discrepancy can create a bias towards the majority classes during training, causing the model to perform poorly when predicting minority classes.

To manage this issue, class weights were used. In essence, class weights apply a higher penalty to the model when it makes an incorrect prediction for a minority class. Consequently, the model is “encouraged” to pay more attention to these less represented classes during training. Weights are usually inversely proportional to the class frequencies. The majority class gets a lower weight, whereas the minority class is assigned a higher weight.

Table 3: Weights of the elements classification dataset

Class	No. of images	Weight
Altar	828	12.23
Apse	505	20.06
Bell Tower	1057	9.58
Column	1913	5.29
Dome (inner)	589	17.20
Dome (outer)	1175	8.62
Flying Buttress	405	25.01
Gargoyle	1562	6.48
Stained Glass	998	10.15
Vault	1097	9.23

In this case, for every incorrect prediction made on class 7, for example, the model “loses” roughly 5 times more than it would for an incorrect prediction on class 4, thus enhancing the model’s predictive power particularly on the minority classes.

However, the use of class weights should be done cautiously. Assigning too much importance to the minority classes can lead to overfitting specifically to these classes. This can consequently hinder the performance on majority classes and on unseen data. Therefore, finding the optimal class weights often requires some iterations and tuning.

3.2 - Object detector

Above is a concise description of how one can structure a convolutional neural network for image classification. As the goal of this project was not only to identify whether a particular object exists in an image, but also to discern its location and scale and do this for multiple objects in the same image, a standard image classification network did not suffice. This is the case because what was developed simply takes an image and assigns it to a class. The incorporation of methods that could help spatially locate objects within images and handle object scales was in order, leading to the implementation of the sliding window and image pyramid techniques into the model.

Object detection works by localizing instances of objects of a certain class within an image, adding significant value and depth of insight to the model, as well as giving it utility to real-world applications. The addition of an object detector levels up the previous Convolutional Neural Network to not only classify, but also accurately determine where an object resides within an image and at what scale.

3.2.1 - Sliding window

The sliding window approach involves moving a window, a region of interest (ROI), across the image and running the classifier on this portion. The goal is to determine whether a given object is present in this window or not. It will slide across all areas of the image, allowing the model to detect the object in different locations, essentially assigning it to a class.

To ensure that this has the expected outcome, some measures must be taken, such as setting some parameters. Setting the dimension of the ROI to be the same as the dimension of the input layer of the model was the first step. This parameter designates the fixed size to which each window is resized before being fed into the CNN. The next parameter is the step size in both the x and y axis of the sliding window. The smaller this is, the higher the overlap between consecutive windows, and the more windows need to be processed, which greatly increases computation time, but can potentially lead to better detection performance.

Each region that is created is then inputted into the CNN, meaning that every single one will have an assigned class. Even if there is no object of interest, it will simply have something like a 10% probability for each class. To counteract this, a parameter of minimum confidence was defined. It is the minimum probability required for a positive prediction. Spots with a probability lower than this threshold will be ignored.

However, only using sliding windows could turn out to be computationally demanding and may not properly handle objects of different sizes and orientations. This is where the image pyramid technique comes into play.

3.2.2 - Image pyramid

Image pyramiding involves creating a series of images that are scaled-down versions of the original image (Figure 8). It is basically like viewing the image at various resolutions. It starts with the original image size and progressively decreases it, creating a group of images of the same object at different scales. By applying this approach at each level of the image pyramid, the algorithm can detect objects at various scales even though the sliding window is of a fixed size.

As before, to ensure optimal performance and accuracy, the setting of some parameters is key. The scale and downscale variables control how much the width and height of the image are reduced at each layer and step, respectively. A lower downscale value will result in more levels in the pyramid, allowing the detector to identify smaller objects, but greatly increasing the amount of computation required. To prevent the creation of extremely small images, which might create issues during the classification process, a minimum size parameter is defined, which is the dimension of the smallest images at the lowest level of the pyramid.

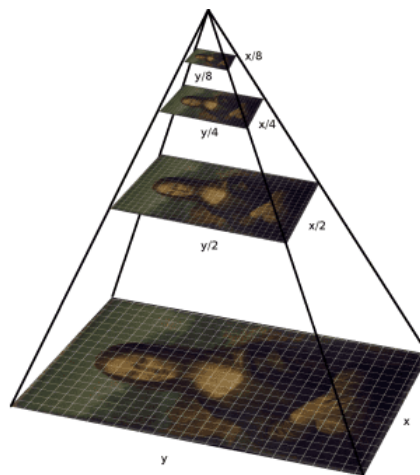


Figure 8: Image pyramid (Source: Web)

With the combination of the sliding window and image pyramid techniques, the model was allowed to perform object detection, in an accurate and versatile way, in terms of both the position and size of the objects present in the image.

3.3 - Conclusion

In conclusion, this chapter has described the core methodologies employed within this dissertation, providing a robust foundation for the development of and application of Deep Learning techniques tailored specifically to image analysis in the sphere of cultural heritage.

Highlighting the more original and crucial aspects of the adopted strategy, this work is positioned within existing theory and practice, thus innovating by introducing a new analytical landscape to the larger field.

The strategies discussed have not only enhanced the existing principles used in cultural heritage accessibility but have also paved the way for more comprehensive and targeted applications of Deep Learning methodologies in the future.

The next section will assert the processes of experimentation and the consequential results obtained from the deployment of the discussed techniques, further shedding light upon the premise and potential of this approach.

Finally, it is important to note that, in parallel, a very similar approach was followed for the development of a separate model to classify a building's architectural style. The intricate complexities of various architectural styles were fed into this model, empowering it to differentiate and categorize different movements. This subsequent initiative further amplifies the scope of this research, reaffirming the goal of using deep learning techniques to explore and better document the world's rich and diverse cultural heritage.

4 Results

This chapter presents the key findings derived from the implemented processes detailed in the preceding sections, recapping the main research goals and the techniques carried out. Along the way, a framework for the application of Deep Learning techniques to the context of cultural heritage was developed. Given the scarcity of previous research in marrying these two domains, the results obtained hold promise for further exploration and affirmation of this intersection of technology and culture.

The central point of this research resides in the results generated from the meticulous application of the model developed. The product of this approach manifests itself in the final image that the model rendered.

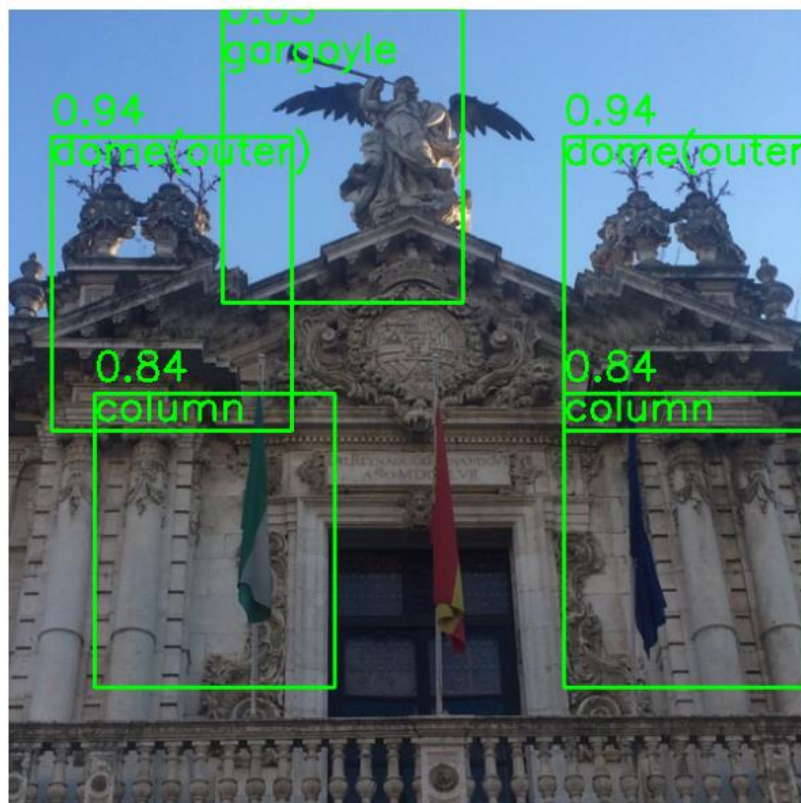


Figure 9: Result of the object detector

Illustrated in Figure 9, this image encapsulates three of the ten diverse elements the model was built to identify, correctly classifying and locating them in a never seen image. Although not perfect, it achieved a very significant accuracy for the represented classes (83% for gargoyle, 94% for dome, and 84% for column), and a reasonably good ability to pinpoint their exact location.

Furthermore, the second model was able to very accurately classify an image's architectural style, as portrayed by its output in the next figure.

This image most likely belongs to Baroque with a 91% percent confidence.

Figure 10: Result of architectural style classification

Diving deeper into the examination, development, and training of the model used, getting a closer look at its architecture is fundamental to understanding the foundation of the results produced. This model's design is layered scheme, where each layer performs a specific task that sequentially contributes to the final output.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 64, 64, 3)]	0
sequential (Sequential)	(None, 64, 64, 3)	0
resnet50 (Functional)	(None, None, None, 2048)	23587712
rescaling (Rescaling)	(None, 2, 2, 2048)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
dense (Dense)	(None, 512)	1049088
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8256
dropout_3 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 10)	650
Total params: 24,809,930		
Trainable params: 4,637,770		
Non-trainable params: 20,172,160		

Figure 11: Model summary

As seen in Figure 11, the resnet50 layer represents the imported resnet model, where only 10 layers were allowed to be trained, resulting in the 23587712 trainable parameters, which proved to be the optimal number through various iterations, followed by the custom part of the final network to allow for our case study of 10 classes. The addition of the first layer was also needed to allow these study's custom inputs.

As stated, the process of finding the right values is centered around an extensive iteration of many runs, monitoring some metrics. Through the course of the presentation of the methodology only accuracy and loss were mentioned, but some more metrics must be analyzed in order to be sure the model is performing as intended. The next table illustrates three more metrics that give a better understanding on the model's performance.

Table 4: Additional metrics

Class Name	Precision	Recall	F1-Score	Support
Altar	0.89	0.91	0.90	140
Apse	0.83	0.86	0.84	50
Bell Tower	0.94	0.94	0.94	170
Column	0.94	0.91	0.92	210
Dome (inner)	0.84	0.84	0.84	69
Dome (outer)	0.88	0.89	0.89	142
Flying Buttress	0.92	0.99	0.94	70
Gargoyle	0.95	0.92	0.94	240
Stained Glass	0.91	0.91	0.91	150
Vault	0.94	0.91	0.92	163

Precision is the ability of the classifier not to label a negative sample as positive. In other words, it is the ratio of true positives to the sum of true positives and false positives (the total actual positives).

Recall (also known as Sensitivity, Hit Rate, or True Positive Rate) is the ability of the classifier to find all the positive samples correctly. It is the ratio of true positives to the sum of true positives and false negatives (the total predicted positives).

F1-score is a harmonic mean of precision and recall. A value closer to 1 indicates a better F1 score, while a value toward 0 indicates a worse score.

Support is simply the number of samples of the true responses that lie in this class.

Other than this, a common way to assess the correct implementation of a model is the confusion matrix. This is a table layout that allows the visualization of the performance of the algorithm. Each column of the confusion matrix represents the instances in a predicted class

while each row represents the instances in an actual class which, in this case, is of dimensions 10x10.

Table 5: Confusion Matrix

Class	Altar	Apse	Bell Tower	Column	Dome (inner)	Dome (outer)	Flying Buttress	Gargoyle	Stained Glass	Vault
Altar	127	3	2	1	0	0	0	1	1	5
Apse	2	43	2	2	0	0	0	1	0	0
Bell Tower	0	3	160	2	0	0	0	1	2	2
Column	1	2	2	191	0	0	0	1	0	13
Dome (inner)	0	0	0	0	58	3	0	0	5	3
Dome (outer)	0	0	0	0	3	126	0	1	2	10
Flying Buttress	0	0	0	0	0	0	69	0	0	1
Gargoyle	0	0	1	0	0	2	0	221	0	16
Stained Glass	0	0	1	0	1	0	0	0	136	12
Vault	2	0	2	10	1	4	0	6	8	148

For example, from the confusion matrix:

The first value 127 at position (0, 0) means there are 127 instances which are of class Altar and the model also predicts them as class Altar.

The second value 2 at position (1, 0) suggests there are 2 instances which are of class Apse, but the model predicts them as class Altar. Analogously, the value 3 at position (0, 1) tells that there are 3 instances which are of class Altar, but the model predicted them as Apse.

In an ideal case, we would like to see high values down the main diagonal (top left to bottom right) of the confusion matrix (indicating many true positives), and zeros everywhere else (indicating no false positives or false negatives).

The precision, recall and f1-score, as well as the confusion matrix were obtained by feeding a collection of never seen images to the model, and tracking its performance. By analyzing these results in parallel to the output image and assigned class, the performance of the model could be inferred over a greater number of iterations, comparing different values of the various hyperparameters to arrive at the optimal solution.

To wrap things up, the final training process consists of 13 epochs, achieving an accuracy of 92.05% and highest validation accuracy of 88.76%. As seen in figure 12, both these accuracies steadily increase, while the training and validation losses steadily decrease, which is exactly what a well architected model should do.

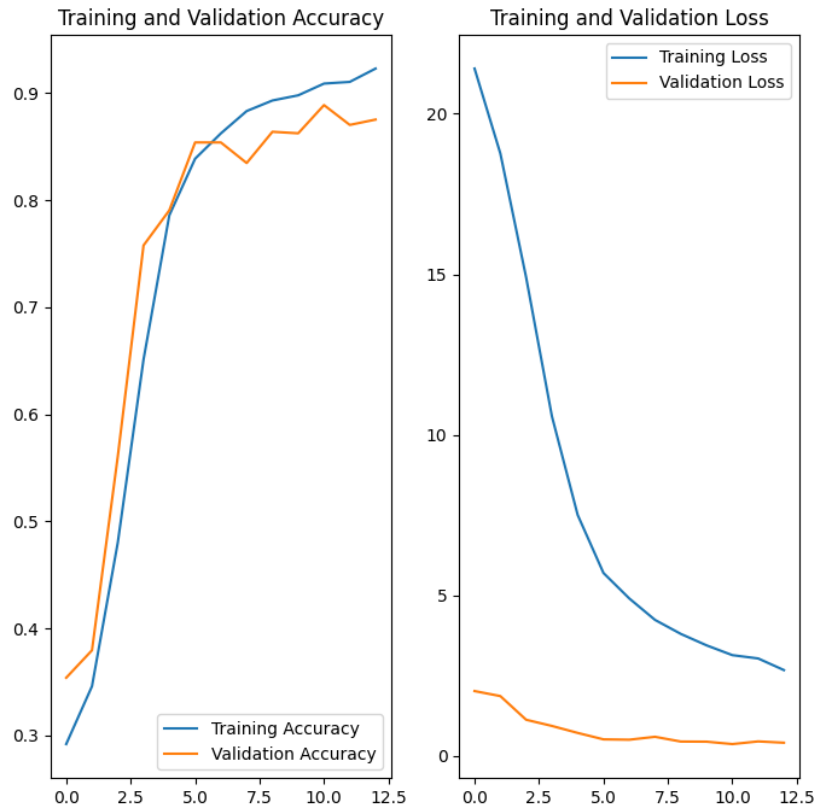


Figure 12: Final accuracy and loss

Understanding this architecture aids in comprehending the complexity of the whole process and the sophistication of the techniques involved. Despite facing many challenges, the model's steady improvement and the ultimate results obtained exemplify the robustness and utility of deep learning techniques in the domain of cultural heritage, and the potential of AI-infused architecture analysis in providing new perspectives.

5 Conclusion and future work

Recognizing limitations forms an integral part of any vigorous research, providing insight for improvement and setting directions for future work.

It is important to note that this thesis implies addressing preservation further. The primary challenge that rose in this regard was the scarcity of labeled datasets suitable for training deep learning models in this context. This deficiency has long been acknowledged as a loophole in this area, significantly limiting the ability to extensively tackle the preservation side of this dissertation.

This constriction, nonetheless, should not be seen as an unsurpassable roadblock but rather as a driving force for future research directions. Acquiring and cultivating rich and labeled datasets for cultural preservation bring along its own complexities and challenges, yet its worth is undeniable. The creation of such datasets could revolutionize how Deep Learning models contribute to cultural heritage preservation. It would truly bring this work and future efforts in the field together into a unified, data-driven analysis and preservation of the world's cultural heritage.

Thus, as things progress, the call is not only for continuous refinement of analytical methods and strategies, but also for tackling these data availability hurdles and advancing towards comprehensive solutions for preservation. This essential work could very well be the future of leveraging Deep Learning technology to its full potential in cultural heritage preservation, restoration, and accessibility.

On a closing thought, the research presented in this paper underscores a world of possibilities. The challenges encountered, the results obtained, and the experience gathered offer an enlightening perspective of this intersection of technology and culture.

While the absence of detailed datasets for this specific context imposed some limitations on this work, it undeniably creates an ample avenue for future research and development. Resolving this data scarcity and creating rich databases will certainly transform the way we think and interact with our cultural heritage.

So, as this thesis suggests, we find ourselves at the doorway of a transformative era in culture preservation. An era potentiated by the immense potential within Artificial Intelligence technology. And as we cross this line, we step into a landscape where a renewed connection to our past shapes the rich cultural legacy we pass on to future generations.

References

- [1] UNESCO Institute for Statistics, 2009 UNESCO Framework for Cultural Statistics
- [2] T. Nilson and K. Thorell, "Cultural Heritage Preservation: The Past, the Present and the Future," 2018
- [3] S. Yazdani Mehr, "Analysis of 19th and 20th Century Conservation Key Theories in Relation to Contemporary Adaptive Reuse of Heritage Buildings," *Heritage*, vol. 2, no. 1, pp. 920-937, Mar. 2019
- [4] H. Mekonnen, Z. Bires, and K. Berhanu, "Practices and Challenges of Cultural Heritage Conservation in Historical and Religious Heritage sites: Evidence from North Shoa Zone, Amhara Region, Ethiopia," *Heritage Science*, vol. 10, no. 1, Oct. 2022
- [5] J. McCarthy, "From here to human-level AI," *Artificial Intelligence*, vol. 171, no. 18, pp. 1174-1182, Dec. 2007
- [6] A. Turing, "Computing machinery and intelligence," *Mind*, vol. LIX, no. 236, pp. 433-460, Oct. 1950
- [7] Stuart J. Russell and Peter Norvig, "Artificial Intelligence A Modern Approach Third Edition."
- [8] OECD, "Artificial Intelligence, Machine Learning and Big Data in Finance Opportunities, Challenges and Implications for Policy Makers," 2021
- [9] S. Ali et al., "Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence," *Information Fusion*, vol. 99, no. 101805, p. 101805, Apr. 2023
- [10] K. Siountri and C.-N. Anagnostopoulos, "The Classification of Cultural Heritage Buildings in Athens Using Deep Learning Techniques," *Heritage*, vol. 6, no. 4, pp. 3673-3705, Apr. 2023
- [11] F. Monna et al., "Deep learning to detect built cultural heritage from satellite imagery. - Spatial distribution and size of vernacular houses in Sumba, Indonesia -," *Journal of Cultural Heritage*, vol. 52, pp. 171-183, Nov. 2021
- [12] Magdalena Pasikowska-Schnass with Young-Shin Lim Members' Research Service, "Artificial intelligence in the context of cultural heritage and museums", PE 747.120 - May 2023
- [13] "3Blue1Brown," www.3blue1brown.com/topics/neural-networks
- [14] Mohammed Hamzah Abed, Muntasir Al-Asfoor, and Zahir M Hussain, "Architectural Heritage Images Classification Using Deep Learning With CNN."
- [15] A. Lamas et al., "MonuMAI: Dataset, deep learning pipeline and citizen science-based app for monumental heritage taxonomy and classification," *Neurocomputing*, vol. 420, pp. 266-280, Jan. 2021
- [16] M. E. Hatir, M. Barstuğan, and İ. Ince, "Deep learning-based weathering type recognition in historical stone monuments," *Journal of Cultural Heritage*, vol. 45, pp. 193-203, Sep. 2020
- [17] "Architectural Heritage Elements Image64 Dataset," <https://www.kaggle.com/datasets/ikobzev/architectural-heritage-elements-image64-dataset>

[18]“ari-dasci/OD-MonuMAI,” GitHub, Aug. 09, 2023. <https://github.com/ari-dasci/OD-MonuMAI>