Automation of the Labeling Process Using an Image Classification Model Using Convolutional Neural Networks

Diego Veliz, Ronald Ccori, and Luis Alfaro *

Universidad Nacional de San Agustin (UNSA), Arequipa, Peru
Email: dvelizs@unsa.edu.pe (D.V.); rccorih@unsa.edu.pe (R.C.); casas@unsa.edu.pe (L.A.)
*Corresponding author

***Abstract*—Emerging technologies enable the refocusing of communication and interaction strategies with clients and users through significant innovations, such as the use of 360° videos and immersive Virtual Reality (VR) in tourism and hotel promotion. The aim of this work is to leverage these technologies to optimize the creation of 360° experiences through process automation focused on the classification of images that will compose such experiences. In our proposal, we designed a Convolutional Neural Network (CNN), whose essential functions are feature extraction and image classification and output processes, as these will be used for the composition of virtual tours. The feature extraction stage consists of several hidden layers, such as the convolution layer, the Rectified Linear Unit (ReLU) activation function, and the pooling layer. Subsequently, training and testing are conducted to ensure that the labeling process of 360° videos is automated by the virtual tour viewer prototype, optimizing a process traditionally performed manually. The model's functionalities and test results were satisfactory, achieving 95.09% accuracy, surpassing the success indicators for such a model. Finally, conclusions and recommendations for future work are established.**

***Keywords*—machine learning, convolutional neural network, image classification, labeled**

I. Introduction

Currently, technologies have become essential for humans in various aspects of life. They facilitate the management and exchange of information globally and efficiently, benefiting multiple fields.

One such field is machine learning, the scientific study of algorithms and statistical models that use computer systems to perform specific tasks without explicit programming [1]. Its use offers significant advantages and optimizes processes that would otherwise take a long time to execute. Additionally, there are user-friendly tools available to maximize the performance of machine learning models.

There are various machine learning models available today that can be applied to a range of problems or situations, depending on the amount of data being handled, the model best suited to the problem, the type of solution desired, and other factors.

Convolutional Neural Networks (CNNs) are a type of neural network primarily used for classification tasks and computer vision. They leverage principles of linear algebra, specifically operations such as matrix multiplication, to identify patterns in an image.

This pattern recognition is an excellent tool that enables the automation of multimedia classification processes, a task that would require a significant amount of time if done manually. A well-trained model significantly speeds up this process. Its utility ranges from enhancing images and videos in general to detecting tumors in medicine or performing facial recognition [2].

This work focuses on optimizing a traditionally manual process by transforming it into an automated process using an image classification model based on a convolutional neural network. The article is structured as follows: Section II details the literature review, Section III presents related work, Section IV outlines the proposed methodology for the development of the software prototype, Section V discusses the testing and results, and finally, Section VI provides conclusions and recommendations for future work.

II. State of the Art

A. Machine Learning

The philosophy behind Machine Learning (ML) involves the automatic creation of analytical models, enabling algorithms to continuously learn from available data. The use of machine learning techniques is increasingly prevalent due to the vast amounts of complex data available and the ease of use these techniques offer [3].

Machine learning has progressively advanced, starting from the theory of computation and pattern recognition. Currently, there are several tools available to implement machine learning-based solutions, but not all of them can adapt to the variety of possible applications and scenarios [4].

Machine learning is based on various algorithms to address data-related problems. Scientists often emphasize

Manuscript received March 1, 2024; revised May 21, 2024; accepted July 8, 2024; published September 13, 2024.

that there is no single algorithm that is universally optimal for solving all problems. The choice of algorithm depends on the specific problem to be addressed, the number of variables involved, the type of model that best fits the data, and other contextual factors [1].

B. Deep Learning

Deep learning, also known as deep neural networks, is a subset of machine learning characterized by neural networks with a large number of layers and parameters. Most deep learning methods utilize neural network architectures.

Deep learning employs a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. The lower layers, close to the data input, learn simple features, while the upper layers learn more complex features derived from the features of the lower layers. This architecture forms a powerful hierarchical representation of features [5].

C. Convolutional Neural Network

The neural network model consists of a training phase, in which the network's parameters are estimated from a given training dataset, and a testing phase, which is applied to the trained network to predict the classes of new input data [6].

The Convolutional Neural Network (CNN) is a type of Artificial Neural Network with supervised learning, whose structure is hierarchically organized through layers. These layers mimic the visual cortex of the human eye to learn to extract the most representative visual features of an image, with the aim of using them to achieve a specific objective [7].

Its ability to manage objects can be controlled by varying its depth and breadth. CNNs also make robust and mostly correct assumptions about the nature of images. Consequently, compared to standard neural networks with layers of similar size, CNNs have significantly fewer connections and parameters, allowing for better and more straightforward training [8].

The CNN consists of a series of hidden layers specialized in processing and identification. The initial layers can detect lines and polygonal shapes, while deeper layers can recognize complex figures such as faces or silhouettes. This model takes as input the dimensions of the images as well as the colors, which represent the channels.

Despite the development of new techniques and technologies, convolutional neural networks remain a fundamental tool due to their ability to extract spatial features and patterns in image data.

In Table I, a comparison of CNNs against other machine learning techniques is presented. SVM (Support Vector Machine) demonstrates high efficacy in image and text classification tasks; however, CNNs surpass them [9, 10]. SVMs are also inefficient in object detection and image segmentation due to their limited ability to learn complex spatial features [11].

On the other hand, KNN (K-Nearest Neighbors), an instance-based learning method, shows acceptable efficacy in image and text classification tasks [9, 10], but it is less effective in complex tasks due to its inability to effectively utilize complex spatial features [11].

TABLE I. COMPARATIVE TABLE OF LEARNING TECHNIQUES EFFICIENCY

Task	Metrics	CNN	SVM	K-NN	Classic Neuronal Networks
Image classification	Accuracy	98%	86%	82%	92%
Detection of objects	mAP (Pascal VOC 2007)	76%	42%	34%	60%
Facial Recognition	Rate of error	0.3%	2.3%	3.7%	1.7%
Image Segmentation	IoU(Pascal VOC 2012)	79%	47%	40%	62%
Text Classification	Accuracy	95%	89%	87%	90%

Classical neural networks, in contrast, exhibit reasonable efficacy across various tasks but not as much as CNNs due to their lack of convolutional operations [9, 12]. Another reason for their lower efficacy is their susceptibility to overfitting in high-dimensional tasks such as image processing [13].

Continuing, we detail the layers comprising the convolutional neural network architecture and their operational principles.

1) Convolutional layers

The convolutional layers are the primary component of CNNs, employing filters (kernels) that traverse the input image to extract local features and patterns through convolution operations [2]. This is exemplified in EfficientNet, where Tan and Le [14] introduced a compound scaling method that optimizes the depth, width, and resolution of convolutional layers, resulting in significant improvements in CNN performance and efficiency.

2) Activation layers

The activation layer applies non-linear functions within the model, allowing it to capture non-linear relationships in the data, which are essential for more complex classification tasks. The most common activation function is Rectified Linear Unit (ReLU), which helps mitigate the vanishing gradient problem, thereby facilitating the training of deep networks. Additionally, ReLU has variants such as Leaky ReLU and ELU, which introduce improvements for better gradient flow management and enhanced learning capacity [2].

3) Pooling layers

The pooling layers, such as Max Pooling and Average Pooling, reduce the dimensionality of feature maps while retaining the most important features and decreasing sensitivity to translations [2].

The benefits of Max Pooling include preventing the model from learning incorrect details or noise in the training data, improving the model's robustness, preserving important features, and optimizing computational efficiency [15, 16].

4) Fully connected layers

Fully connected layers take the final feature map and transform it into an output of fixed dimensions, which is

useful for classification or regression tasks. Each neuron in these layers is connected to all neurons in the previous layer, allowing the network to make a final decision based on all learned features throughout the network [2].

5) Normalization layers

Batch Normalization is used to normalize the inputs of each mini-batch using two main functions: First, it accelerates training by enabling higher learning rates and reducing the need for regularization. Second, it enhances stability by normalizing the output of intermediate layers, thereby mitigating drastic changes in activation distribution [2].

The distinctive procedure of this model revolves around convolutions, where a randomly generated kernel/filter is used to traverse the image, computing dot products to populate an output matrix, which serves as one of the hidden layers. These matrices contain features from the original image and will help identify what the search corresponds to in the future. Following this, the ReLU function is applied through another layer to decrease negative values, and pooling is used to reduce the sample of feature maps before activating the value [17].

To reduce the size of the next layer of neurons, a subsampling process is performed where the size of filtered images is reduced, aiming to emphasize the most important features detected in each filter. One of the most commonly used methods for this is Max-Pooling. Multiple iterations are conducted at this stage without restrictions on repetition frequency.

In the final step, the last hidden layer that underwent subsampling becomes a traditional layer of neurons. A function called SoftMax is applied, connecting it to the final output layer that will have a number of neurons corresponding to the classes being classified.

III. Related Work

Various research studies have been conducted based on neural networks, focusing on applications in the hospitality industry and also for general image classification.

Cuesta-Valiño *et al.* [18] highlights the significance of images across various fields of study, with a particular emphasis on hospitality. It establishes the importance of using neural network models to analyze how properties such as quality, composition, and selection of images contribute to user attraction. The research confirms that these factors play a crucial role in user engagement.

Similarly, other studies highlight the use of Convolutional Neural Network (CNN) models in the medical field, particularly in the detection of threats in medical analyses and examinations. In Ref. [19], CNN models were employed for the detection of tumors and anomalies, achieving an accuracy close to 99%. In Ref. [20], a hybrid variation of the model, combining Convolutional Neural Networks with Support Vector Machines (CNN-SVM), was utilized for a similar purpose in the classification of circulating tumor cells, achieving successful results with an accuracy rate exceeding 90%.

IV. Methodology of Prototype Development

A. Research Proposal

The work described in [21] involves the development of a prototype for visualizing virtual tours in hotels using Virtual Reality (VR) and Augmented Reality (AR). In the prototype development process, manual processes are employed for labeling 360° videos to be used in the virtual tour functionality.

This manual process includes downloading the video and converting it to a flat video format, typically MP4. Subsequently, the video is viewed to determine which environment it belongs to, and it is then stored with the corresponding environment's name. This labeling process takes approximately 2 min per video, which accumulates to a significant amount of time when labeling videos from one or multiple hotels.

In this article, the objective is to enhance and automate the process described earlier using Convolutional Neural Networks (CNNs). The goal is to optimize the labeling phase of 360° videos, transitioning from a manual process to an automated one.

The proposed method aims to streamline the tagging of 360° videos by leveraging CNNs. Fig. 1 illustrates the flow of this proposal.

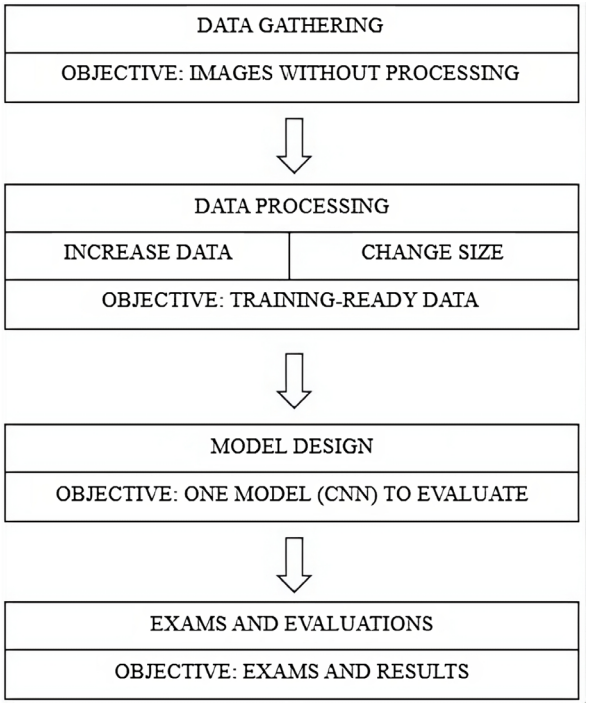

Fig. 1. Proposal flowchart.

B. Data Acquisition

In the proposal, data is collected from images of common hotel environments such as bathrooms, bedrooms, living rooms, patios, and pools. The data was acquired from various internet sources containing datasets of the required images. In total, 500 images per environment were collected, resulting in a total of 2500 images.

After this collection process, the dataset is split into 80% for training and 20% for testing purposes. Fig. 2 illustrates the selected images of the environments to establish the dataset.

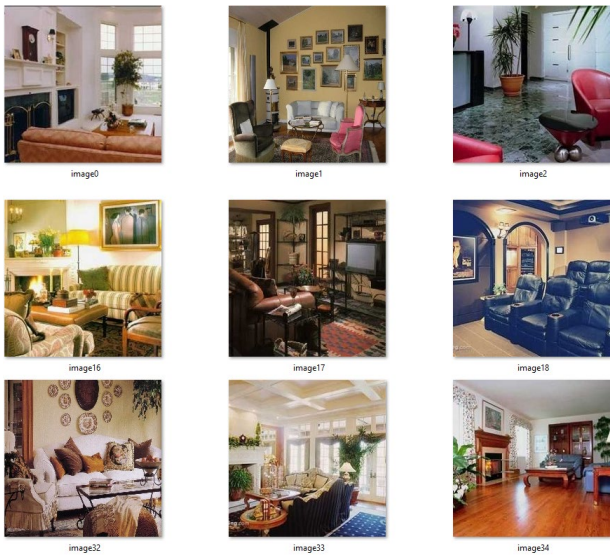

Fig. 2. Collected images.

C. Data Preprocessing

Data preprocessing involves a set of processes carried out on the data to prepare them for use as input in training. In this proposal, the following processes will be performed to prepare the images.

1) Data augmentation

This process involves increasing the data because the collected dataset is limited. To achieve better classification accuracy, the proposal aims to have a minimum of 1000 images per environment. However, obtaining the minimum amount of data manually is challenging. Therefore, the solution is to horizontally flip each image, effectively doubling the number of images for each originally collected image. Fig. 3 provides an example of data augmentation.

Fig. 3. Augmented images.

2) Size change

This involves resizing the images to standardize their size. It is done to equalize the image size for each data point because images obtained from different datasets may come with varying sizes and dimensions. In this proposal, all images will be resized to 224×224 pixels.

D. Model Design

In the proposal, after completing the data preprocessing and preparing it for use, a procedure is undertaken to design and construct a model for the learning process.

A Convolutional Neural Network (CNN) is designed for this purpose, as it is well-suited for processing image inputs. This is primarily because each neuron in the CNN is represented in two dimensions, aligning with the spatial nature of images. The CNN structure comprises an input layer, a feature extraction process, a classification process, and an output layer.

The feature extraction process includes several hidden layers such as convolutional layers, the Rectified Linear Unit (ReLU) activation function, and pooling layers.

The architecture created for the CNN model is shown in Fig. 4, and Fig. 5 shows the details of the configured architecture.

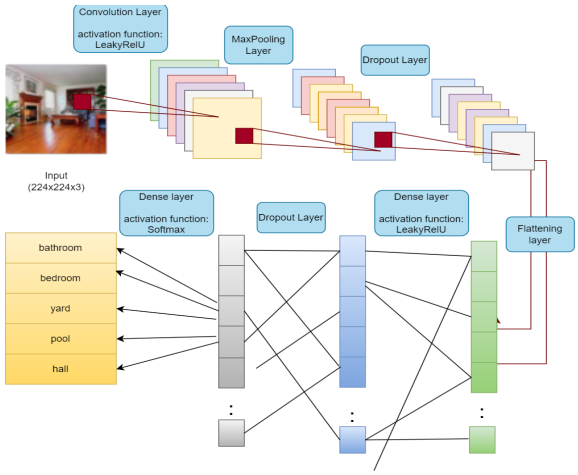

Fig. 4. CNN architecture.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 224, 224, 32)      896

 leaky_re_lu (LeakyReLU)     (None, 224, 224, 32)      0

 max_pooling2d (MaxPooling2  (None, 112, 112, 32)      0
 D)

 dropout (Dropout)           (None, 112, 112, 32)      0

 flatten (Flatten)           (None, 401408)            0

 dense (Dense)               (None, 32)                12845088

 leaky_re_lu_1 (LeakyReLU)   (None, 32)                0

 dropout_1 (Dropout)         (None, 32)                0

 dense_1 (Dense)             (None, 5)                 165

=================================================================
Total params: 12846149 (49.00 MB)
Trainable params: 12846149 (49.00 MB)
Non-trainable params: 0 (0.00 Byte)
```

Fig. 5. Architecture detail.

1. The input image of size (224,224,3) passes through the first convolutional layer.
2. The first convolutional layer (Conv2D) applies 32 filters of size 3×3, producing an output tensor of size (224,224,32). The configuration for this layer is as follows:
    a. Filter: 32
    b. Size of Kernel: (3,3)
    c. Activation: Lineal
    d. Padding: same dimensions in order to maintain the inut dimension.
    e. Input mode: Images of 224×224 pixels with 3 color channels (RGB)

The activation function used in this layer is LeakyReLU, a variant of ReLU that allows a small gradient when the unit is not active, addressing the issue of dying neurons. The configuration for this function was: Alpha: 0.1, which defines the slope of activation for negative values (introducing non-linearity).

3. The next layer is the pooling layer, specifically MaxPooling2D, which reduces the dimensionality of the feature maps by selecting the maximum value within each 2×2 window. This helps decrease computational cost and the number of parameters.

In the model design, this layer reduces the tensor size to (112,112,32). The configuration of the layer is as follows:

    a. Pooling size: (2,2)
    b. Padding: Same dimensions to maintain input dimensions.

4. Also utilized is a layer called Dropout, a regularization technique that helps prevent overfitting by randomly deactivating a percentage of neurons during training. In the model configuration, Dropout is set to randomly deactivate 50% of the neurons.
5. Flatten is employed in the flattening layer, which transforms the input. This layer has no parameters to learn, thus it does not alter the input shape. However, it collapses all dimensions into a single entry, except for the first. In the model configuration, Flatten converts the 4D Tensor into a 1D vector of size 401,408 (112,112,32).
6. The next layer is Dense, which are traditional layers in a neural network where each neuron is connected to all neurons in the previous layer.

This layer reduces the vector to 32 dimensions. LeakyReLU activation function is also used in this layer to introduce non-linearity after the dense layer.

Additionally, a Dropout layer is employed to prevent overfitting in the dense layer, deactivating 50% of the neurons.

7. Finally, there is the Output layer, which is a Dense layer with 5 units, where each unit represents a different class.

The Softmax activation function is used in this layer to convert the outputs into probabilities, ensuring that the sum of the outputs is 1. This produces probabilities indicating the likelihood of belonging to each class.

After defining the structure and designing the model, training begins with 100 iterations to achieve improved accuracy.

Finally, once the model is trained, it is saved for use in the labeling phase of developing the virtual tour viewer.

E. Labelling

For the labeling phase, the trained model is used to automate the process. The operation flow of the proposed method is detailed in Fig. 6.

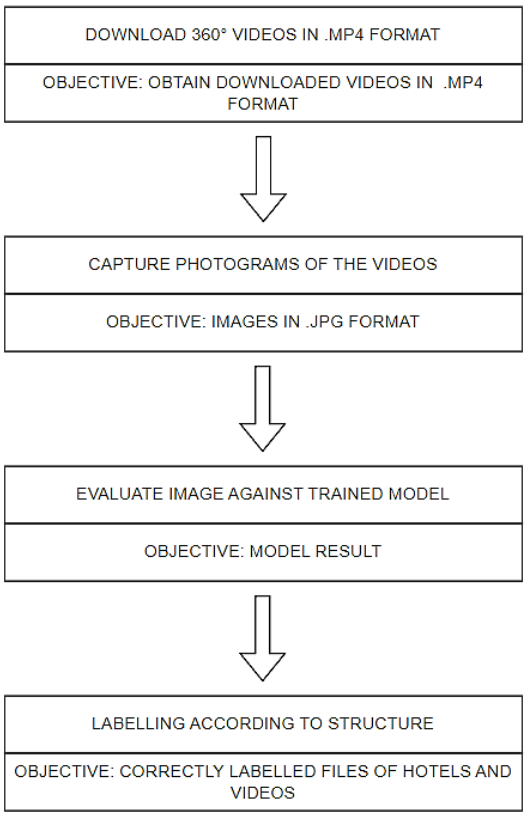

Fig. 6. Labelling flowchart.

1. Download the 360° videos stored in the repository. For this purpose, a program is created to automatically download them.
2. Extract one frame from each video. The aim of this step is to have images in jpg format. The frame extracted from each video undergoes a cropping process to better focus on the environment.
3. With the prepared image, we pass it through the trained image classification model to obtain the name of the environment.
4. Having obtained the name of the environment from the image, we proceed to save the previously downloaded video, but this time with the environment name obtained from the model.
5. We obtain a structure of the stored videos with their respective environment labels, transitioning from a manual process as done previously to an automated one, improving and optimizing the process time.

After having the folder structure and videos correctly labeled, the following steps of the virtual tour viewer prototype development detailed in [15] are carried out.

V. Results and Discussions

Once the neural network was completed, training was conducted with 200 iterations to ensure high accuracy, achieving a precision of 95.09%. Table II illustrats the increase in accuracy across iterations.

TABLE II. Iteration Table

Iteration	Accuracy
1	0.2531
2	0.3772
3	0.4516
97	0.9059
98	0.9056
99	0.9084
100	0.9134
198	0.9519
199	0.9494
200	0.9509

Additionally, in Figs. 7 and 8, the results of accuracy in relation to validation and the loss result in relation to validation are plotted, confirming that the proposed CNN model, using the input images over 200 iterations, achieved a good outcome.

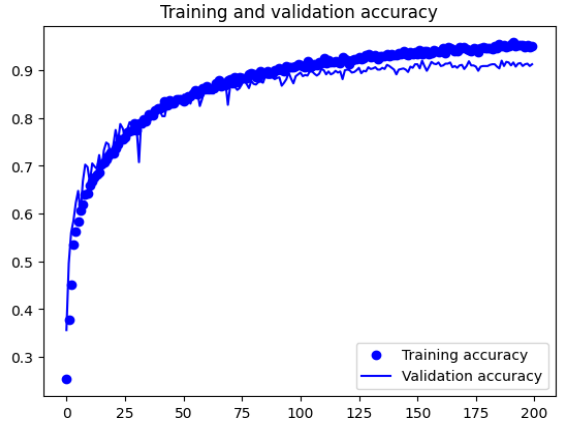

Fig. 7. Accuracy and validation accuracy graph.

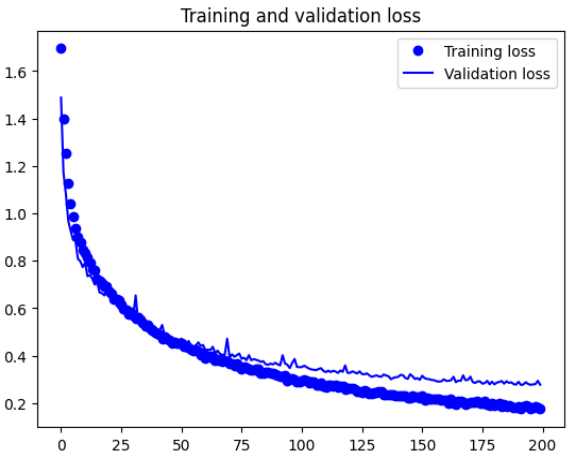

Fig. 8. Loss and validation loss graph.

An overall evaluation of the model was also performed after training using the model.evaluate function, obtaining a loss of 0.2654 and a precision of 0.9160 as shown in Fig. 9.

```
test_eval = ambiente_model.evaluate(test_X, test_Y_one_hot, verbose=1)
32/32 [==============================] - 2s 72ms/step - loss: 0.2654 - accuracy: 0.9160
```

Fig. 9. Evaluation of model.

As part of the evaluation of the results obtained, a verification of the accuracy achieved by the model is also conducted, as shown in Fig. 10. It demonstrates that out of 1000 processed images, 916 were validated correctly, while 84 images had errors, as depicted in Fig. 11. This is primarily due to similarities between rooms, for example, an image of a living room may resemble a bedroom.

Fig. 12 shows the metrics report of the trained model, where it can be observed that Class 4 (Living Room) is the one that obtained the lowest average score in relation to the 3 metrics in the report, with an accuracy of 0.87 and precision of 0.78, along with an F1-score of 0.83, which is a measure that combines precision and recall. It is typically described as the harmonic mean of the two.

Fig. 10. Correctly processed images.

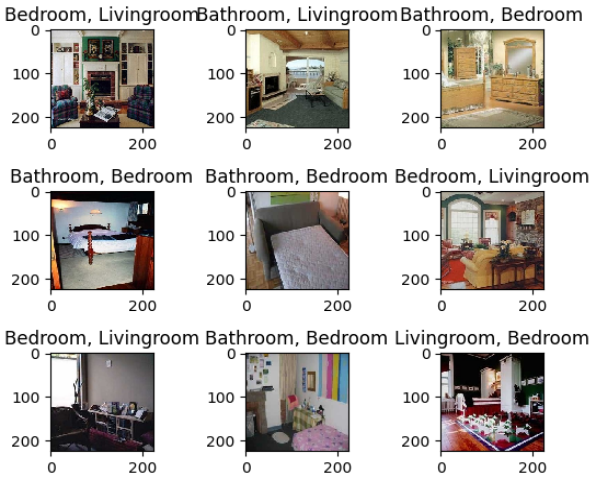

Fig. 11. Incorrectly processed images.

	precision	recall	f1-score	support
Class 0	0.93	1.00	0.97	194
Class 1	0.85	0.86	0.85	192
Class 2	0.92	0.99	0.95	215
Class 3	1.00	0.93	0.96	214
Class 4	0.87	0.78	0.83	185

Fig. 12. Model metrics.

To better understand the results obtained from training the model, a confusion matrix is presented in Fig. 13. Additionally, Table III describes each index for better comprehension.

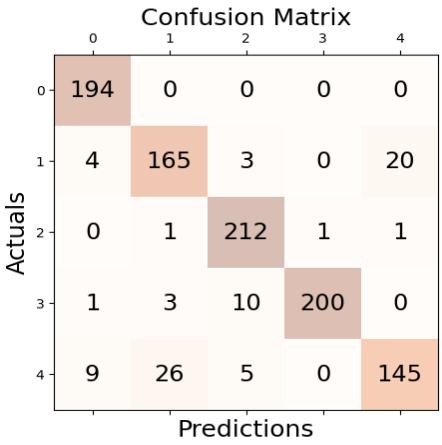

Fig. 13. Confusion matrix.

TABLE III. TABLE OF INDICES BY ENVIRONMENT

Index	Environment
0	Bathroom
1	Bedroom
2	Patio
3	Pool
4	Livingroom
0	Bathroom

According to the confusion matrix, the room with the best precision result is the “Patio” with a value of 212, with only 3 errors in prediction, 1 in “Bedroom”,1 in “Pool” and 1 in “Livingroom.” On the other hand, “Livingroom” had the lowest precision with a value of 145, also having errors in predicting other rooms such as “Bathroom,” “Bedroom,” and “Patio”, with error values of 9, 26, and 5, respectively.

With the result of the model, the labeling process is applied, downloading the videos and extracting a frame in each of them, to which its respective crop is made to pass through the model. The structure and evidence of this procedure is shown in Figs. 14 and 15.

Please improve the clarity of this figure.

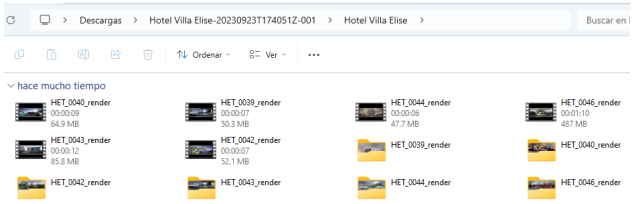

Fig. 14. File with downloaded videos.

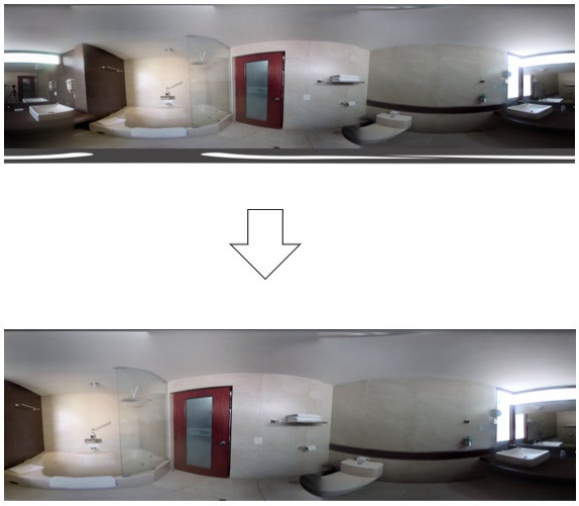

Fig. 15. Extracted image and changed in size.

Finally, based on the result of the processed image through the classification model, which was positive as observed in Fig. 16, the video is stored. The previously downloaded video is now saved with the environment name obtained from the image classification model, resulting in a structure of stored videos with their respective environment labels, as shown in Fig. 17.

```
1/1 [==============================] - 0s 20ms/step
Documents/test/HET_0039_render.jpg Bano
```

Fig. 16. Image of the bathroom correctly recognized by the model.

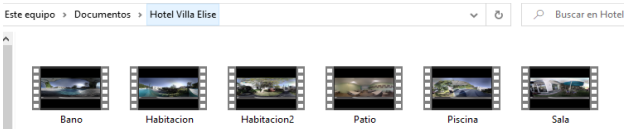

Fig. 17. Properly labeled folder structure.

Based on the results obtained, it can be affirmed that the procedure automates the labeling phase, improving processing time through the application of a properly trained convolutional neural network model. The minimum acceptable precision of the CNN will depend on various specific factors of the context, such as the expected quality by users, the impact of errors on user experience, and industry standards in hospitality [16].

For a hotel virtual tour application, a precision level between 90–95% would be considered acceptable. This value ensures that the majority of selected images are highly relevant and appealing to users [5].

VI. CONCLUSIONS

In this work, we propose a design for an image classification model based on Convolutional Neural Networks (CNNs), achieving positive results for the objective of automating the labeling process of 360° video prototypes in a virtual tour viewer. The success of the CNN model outlined in our proposal automates and optimizes a previously manual process by effectively extracting hierarchical and relevant features from images. This

capability translates into improved identification of scenes and key objects necessary for constructing high-quality virtual tours.

The integration of images selected by convolutional neural networks in virtual tour applications enhances user experience by ensuring that the most relevant and highest-quality scenes are used in the virtual environment. Building on the results obtained in this work, methods can be developed for real-time image selection, enabling dynamic creation of virtual tours. This involves improving the speed and efficiency of CNNs to operate on devices with limited resources, such as VR headsets.

Furthermore, there is potential to scale up to handle large volumes of images, contingent upon factors such as model architecture, hardware infrastructure, optimization techniques, and parallelization approaches. Research efforts could also explore integrating CNNs with other types of neural networks and AI techniques that process multimedia data, enriching the virtual tour experience and providing a completer and more immersive user experience.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Diego Veliz and Ronald Ccori performed the research and data analysis; Luis Alfaro led and guided the research; all three authors participated in writing the paper; all authors had approved the final version.

Funding

To Universidad Nacional de San Agustín de Arequipa (UNSA), for the financing of the research project according to contract No. IBA-IB-12-2020-UNSA.

References

[1] B. Mahesh, "Machine learning algorithms—A review," *International Journal of Science and Research (IJSR)*, vol. 9, no. 1, pp. 381–386, 2020. doi: 10.21275/ART20203995

[2] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "Deep learning for computer vision: A brief review," *IEEE Access*, vol. 8, pp. 187380–187429, 2020. doi: 10.1109/ACCESS.2020.3010147

[3] B. Qolomany, A. Al-Fuqaha, A. Gupta *et al.*, "Leveraging machine learning and big data for smart buildings: A comprehensive survey," *IEEE Access*, vol. 7, pp. 90316–90356, 2019.

[4] I. Fatima, M. Fahim, Y.-K. Lee, and S. Lee, "Analysis and effects of smart home dataset characteristics for daily life activity recognition," *The Journal of Supercomputing*, vol. 66, pp. 760–780, 2013. doi 10.1007/s11227-013-0978-8

[5] P. P. Shinde and S. Shah, "A review of machine learning and deep learning applications," in *Proc. 2018 Fourth International Conference on* Computing *Communication Control and Automation (ICCUBEA)*, Pune, India, 2018, pp. 1–6. doi: 10.1109/ICCUBEA.2018.8697857

[6] C. Cava, S. D. Antona, F. Maselli *et al.*, "From genetic correlations of Alzheimer's disease to classification with artificial neural network models," *Funct. Integr. Genomics*, vol. 293, 2023. doi: 10.1007/s10142-023-01228-4

[7] I. Rodriguez-Martinez, P. Ursua-Medrano, J. Fernandez, Z. Takáč, and H. Bustince, "A study on the suitability of different pooling operators for convolutional neural networks in the prediction of COVID-19 through chest x-ray image analysis," *Expert Systems with Applications*, vol. 235, 121162, 2024.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. the 25th International Conference on Neural Information Processing Systems (NIPS'12)*, 2012, vol. 1, pp. 1097–1105.

[10] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP*, 2014, pp. 1746–1751.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, vol. 1, pp. 886–893.

[12] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3642–3649.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint, arXiv:1301.3781, 2013.

[14] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. the 36th International Conference on Machine*, 2020, pp. 6105–6114.

[15] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*, Cambridge, MA: MIT Press, 2016.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[17] R. Hassan, Z. Sultani, and B. Dhannoon, "Content-based image retrieval based on corel dataset using deep learning," *International Journal of Artificial Intelligence (IJ-AI)*, vol. 12, no. 4, pp. 1854–1863, 2023. doi: 10.11591/ijai.v12.i4.pp1854-1863

[18] P. Cuesta-Valiño, S. Kazakov, P. Gutiérrez-Rodríguez *et al.*, "The effects of the aesthetics and composition of hotels' digital photo images on online booking decisions," *Humanit. Soc. Sci. Commun.*, vol. 10, no. 59, 2023. https://doi.org/10.1057/s41599-023-01529-w

[19] V. E. Papageorgiou, P. Dogoulis, and D. P. Papageorgiou, "A convolutional neural network of low complexity for tumor anomaly detection," in *Proc. Eighth International Congress on Information and Communication Technology*, Springer, 2024, pp. 973–983.

[20] J. Park, S. M. Ha, J. Kim, J.-W. Song, K.-A. Hyun, T. Kamiya, and H.-I. Jung, "Classification of circulating tumor cell clusters by morphological characteristics using convolutional neural network-support vector machine," *Sensors and Actuators B: Chemical*, vol. 401, 134896, 2024.

[21] D. Veliz, R. Ccori, and L. Alfaro, "Division and composition of 360° videos for the generation of personalized tours for tourism, hotel industry and education based on immersive VR," in *Proc. 2023 IEEE World Engineering Education Conference (EDUNINE)*, 2023, pp. 1–6. doi: 10.1109/EDUNINE57531.2023.10102866