Image processing with Artificial Intelligence: coin identifier and counter

Juan Esteban Duque Taborda[1], Juan Camilo Bernal Marín[2], Juan Pablo Uribe[3]

5 de mayo de 2024

Resumen

...

1. Introduction

1.1. Context

Today, Artificial Intelligence (AI) is radically transforming many aspects of our lives, from healthcare to transportation to education. In this context, the need arises to address complex, repetitive and/or automatable problems so that humans can focus on what is necessary. To do this, AI capabilities must be harnessed to deliver innovative and efficient solutions. The university environment is no exception, there are activities that can be automated with artificial intelligence, this is why we decided to lead the project down this path. We decided to implement a solution that would be useful for students, and more specifically for entrepreneurs in the educational field.

1.2. Description of the problem

A very common and often tedious task in our businesses is the accurate identification and counting of coins in various quantities. This problem not only applies to food and snack businesses, but also to jewelry, soft drinks and even collectible figurines. This task consumes entrepreneurs' time and resources that could be invested in other activities, as well as presenting additional challenges when there is high demand and volume, such as miscounting coins or even taking more than one coin at a time.

1.3. Why is it interesting?

The implementation of an AI system capable of identifying and counting coins offers many advantages such as automation and efficiency in cash management. This system could reduce human errors and minimize the risks associated with handling large amounts of cash. In addition, it would provide a scalable and adaptable solution to diverse applications and environments, which could significantly improve productivity, security, accuracy and efficiency in a wide variety of contexts.

2. Problem statement

For the approach of this problem we had to answer different questions that led us to this approach. The questions we posed in a general context for the problem statement were: What activities can be automated with AI? Which people would benefit? In what area would the proposal be applied? Now, we took these questions and modified them in such a way that they would try to answer problems that university students have. What university activities can be automated with AI? What university environments would benefit? In what aspect of university students would the proposal be useful? After having posed these questions, we decided to focus the project on university students' endeavors, where we highlighted the problem of coin identification and counting, which is a problem that resides in any university endeavor.

2.1. Questions of Interest

With respect to questions concerning machine learning, we have the following:

- Is the methodology clear and robust?

- Does the work demonstrate the development of the competencies defined for this course?

3. Theory

To understand which path the project will follow to achieve the solution, we must first adjust the context of the problem to the issues associated with artificial intelligence.

3.1. Type of Problem

In the case of machine learning we have two types of problem, the regression problem and the classification problem [1]. Regression problems seek to predict a continuous numerical value with respect to a set of input variables. These are divided into linear regression, logistic regression, Ridge and Lasso regression, among others. On the other hand, classification problems seek to assign a label or category to an object based on its characteristics. There is binary, multi-class, multi-label and ordinal classification [1]. This is where our problem comes in, since we want to **identify** the coins in an image, so it is a classification problem. Now, which of all the types of sorting problems? The coin identification and counting problem would be a multi-label classification problem, this because in an image there can be more than one coin, and it must not be the same coin, so it is required to pass by parameter more than one label in the same image so that all the coins present can be represented. In other words, the machine learning algorithm must identify and label one or more coins in an image, so it would have to assign multiple classes (labels) to a single data instance (image).

3.2. Categories of Machine Learning

It is also clear to mention that this machine learning algorithm is supervised learning, as the algorithm is given to understand that the data to be learned are labeled in the dataset [7].

Now that we know that the coin identification and counting problem is supervised learning, we must choose which attributes of the data we are going to take for data classification. For coins there are many attributes/features that can be chosen for classification. We choose the most significant characteristics of the coin which are:

1. **coin diameter**: it is relevant because each coin has a different diameter, which can be useful for grading the coin for both new and old.

2. **Thickness of the coin**: it is important in case the image is not totally pointed to the coins, but has some kind of angle, besides it can help us to identify which new coin it is, since new coins change their thickness depending on the price they are worth.

3. **Color of the coin**: it is important because all coins have a unique color, each coin has its own color, no matter if it is new or old, so this characteristic is very relevant when classifying the coin.

4. **Relief and pattern**: this is important, since the figure or symbol on the coin will tell us which coin it is, as well as whether it is new or old. If there are two coins of 100, the relief and pattern will not tell us if it is new or old, which makes it an important characteristic for the classification between the same coins.

3.3. Database

Once the type of problem has been identified, we show the database to be used for this problem. In this case the database was created by ourselves, using a code capable of finding the coins in an image in real time, this method uses **HoughCircles**, which is based on finding circles with the **Hough** transform [2]. This code works, first it asks how many pictures it is going to take, then the name of the folder, then it takes many pictures to the circle it finds in the camera, then that image to a gray scale, makes a crop of the image to only the circle of size 50x50 pixels, then transforms the matrix to a vector, then applies the method **PCA** that eliminates the redundant information of the vector [3], and so it returns us the image and adds it to the folder with the name entered at the beginning. This code returns 18 folders in total.

3.4. Exploratory Data Analysis

Exploratory Data Analysis, simply referred to as **EDA**, is the step where you understand the data in detail. You understand each variable individually by calculating frequency counts, visualizing the distributions, etc. Also the relationships between the various combinations of the predictor and response variables by creating scatterplots, correlations, etc. EDA is typically part of every machine learning / predictive modeling project, especially with tabular datasets [4].

As mentioned before, we have a total of 18 folders, these being the same classes we are going to use. Each class represents an option within the coin, i.e. face or stamp. This is the reason why there are so many classes in our database, because we have a class for both the old and the new coin face and a class for the seal. Within each class there are approximately 200 photos, this is because the code that was used captured, in some occasions, data that were not relevant at all, such as background circles, among others. As mentioned before, color is a really important attribute to classify each coin, but it can be a very confusing issue when classifying the same coins but different states. That is, the 100 coin has the same color for both old and new, and it is so for each coin; each has a unique color and "style"for both old and new, so it can become a point of confusion for the model. But it is for this reason that these attributes are not the only ones taken into account for grading coins.

4. Methodology

4.1. CRISP-DM

The methodology is an adaptation of CRISP-DM (CRoss Industry Standard Process for Data Mining)

1. **Business understanding:** Determinate the objective and requirement for the project. Necessary to identify the project goals and define the porject plan.

2. **Data Understanding:** Identify, collect and analyse the DataSet. These activities are necessary to know the quality of the DataSet and made a recognition of the information.

3. **Data preparation:** Using the results from Data Undetstanding to Select and clean the DataSet, create new DataSet derived from the selected Data.

4. **Modeling:** Select modeling techniques, generate a selection of the test set, build the model make the assess.

5. **Evaluation:** Finishing the training process and created a test, es necessary to evaluate results and Review process to generate adjustments.

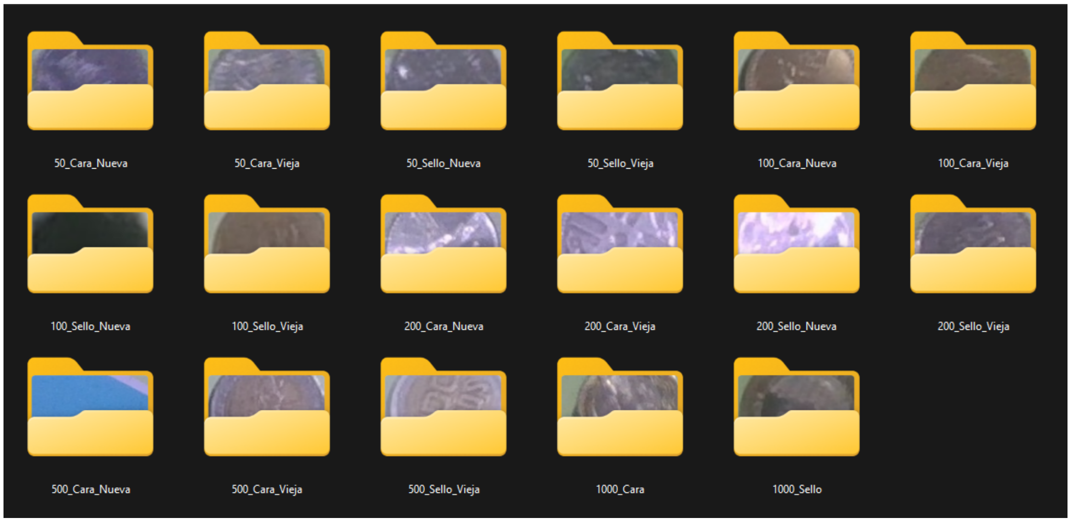
In order to obtain new data, the only thing we did was to retake the same amount of photos for each coin and its type (new or old), just to improve the quality of the image taken, since our data is obtained by ourselves.

As far as the model is concerned, we used the same as explained before, 2 folders per coin with one type, that is to say, 2 folders for the old 100 coin, one folder is face and the other is stamp.

4.2. Application of the models

4.2.1. Preparation

The categories of coins to be used in this study are established by means of: denomination of the coin, face or stamps and time of issue. In the case of the denominations 50, 100, 200, 500 and 1000. On the other hand, it is defined that the time of release corresponds to two categories, new coins that came into circulation in 2012 and old coins, coins in circulation prior to 2012 and of the same denominations. This is reflected in the following image.



Each of these folders is composed of 200 images.

Then, the **Glob** tool is used to obtain the paths of all the images as well as the name of all the categories that are represented in the folder names.

Then, the vector images are obtained, and together with cv2 they are transformed to grayscale.

Finally, reduce the size of each image component and save them in the vector X, necessary to have all the necessary data (images with the above-mentioned processing) for model training.

4.2.2. Model

Separation of the training sets and their respective target values, which have a size of 30 % of the original set X, is performed.

The decision is made to compare two models to select the one with the best accuracy. The first model uses **logistic regression**, after treating the data with **PCA**. The second model is **KNN** using a number of 17 groups.

For the fair comparison it was necessary to train and test with the same selected data sets.

4.3. Deployment plan

If we want to deploy this project so that all people can have access to it, we must follow a plan with which we can give rise, in the best way, to people having a good product, and that the predictions of this model are the best.

1. Data preprocessing: Implements a preprocessing pipeline to standardize the features and apply PCA to reduce the dimensionality of the data.

2. Model training: Train the logistic regression model using the preprocessed data and training set. Adjust hyperparameters as needed, such as the number of principal components in PCA.

3. Model evaluation: Evaluate the model using the test set and evaluation metrics, such as precision, recall, and F1-score. Be sure to validate the model on unseen data to avoid overfitting.

4. Model optimization: If necessary, adjust the hyperparameters of the model and preprocessing process to improve model performance.

5. Production deployment: Deploy the trained model in a production environment, ensuring that it is integrated with the existing system and accessible for use in image-based coin detection.

6. Monitoring and maintenance: Track the performance of the model in production and make adjustments as needed. Keep the model updated with new data and perform periodic testing to ensure its continued effectiveness.

As for the options of where to deploy this model, two were found, which are:

- Web development platforms: If you want to integrate your model into a web application, you can use frameworks such as Flask or Django to develop an API that serves as an interface between the model and end users. This allows you to create an interactive web application where users can upload images and get currency predictions.
- Machine learning automation platforms: These platforms offer complete solutions for the entire machine learning lifecycle, from data preparation to deployment of models in production.

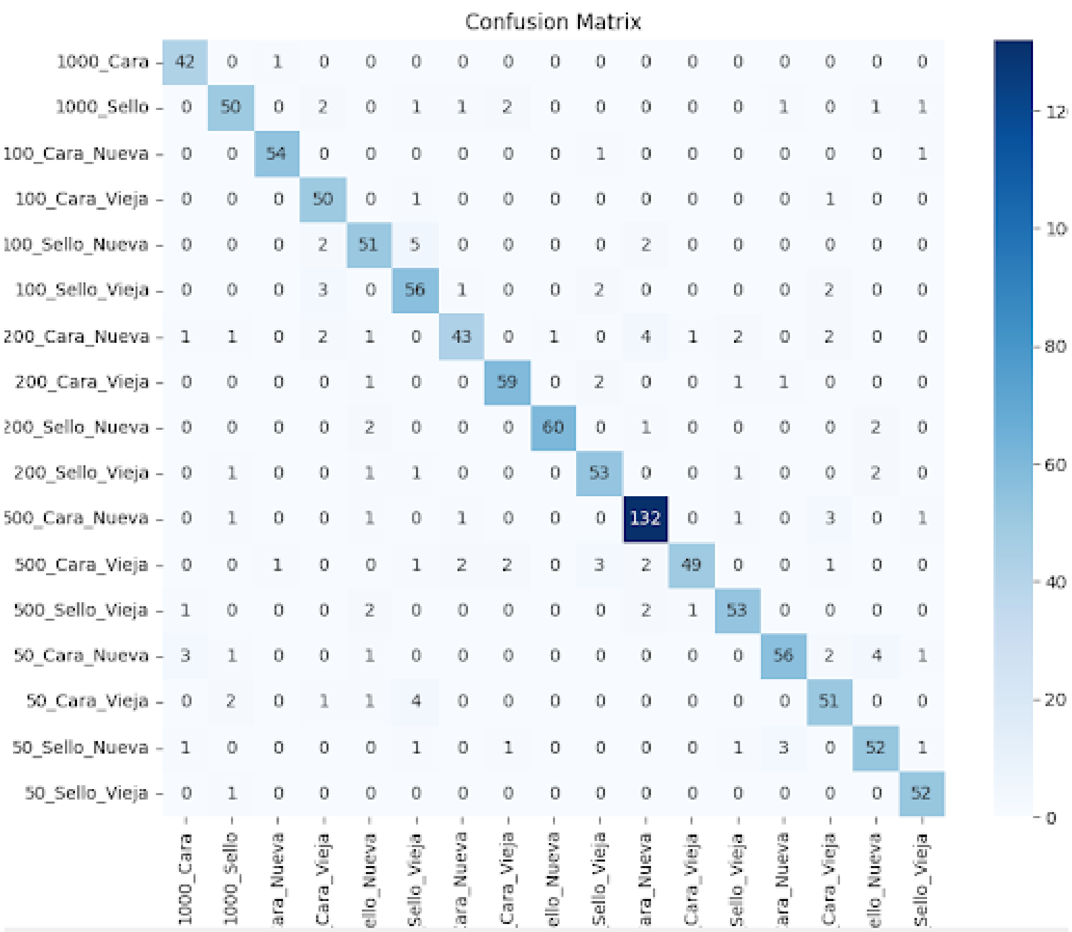
[1] [2] [3]

5. Results

For both algorithms, the confusion matrix was presented to see how much correlation each of the currencies had with the others to see if they are denoted as similar (high correlation) or not (low correlation). This gives us an idea of how difficult the classification will be for each algorithm.

5.1. Logistic regression

For logistic regression we have the following confusion matrix.



[1] Logistic regression is a statistical method used to model the relationship between a binary (or categorical) dependent variable and one or more independent variables. Unlike linear regression, which is used to predict continuous values, logistic regression is used when the dependent variable is discrete in nature and takes only two possible values, such as yes/no, success/failure, or positive/negative.

[2] PCA allows simplifying complex data sets by reducing their dimensionality while preserving as much information as possible. This makes it easier to visualize and understand the underlying structure of the data, as well as to improve the performance of machine learning algorithms by eliminating redundancy and noise in the data.

[3] KNN, or K-Nearest Neighbors, is a machine learning algorithm used for both classification and regression. In this algorithm, given a new data point, the ”k”nearest data points in the training set are identified, where ”kïs a user-defined number.

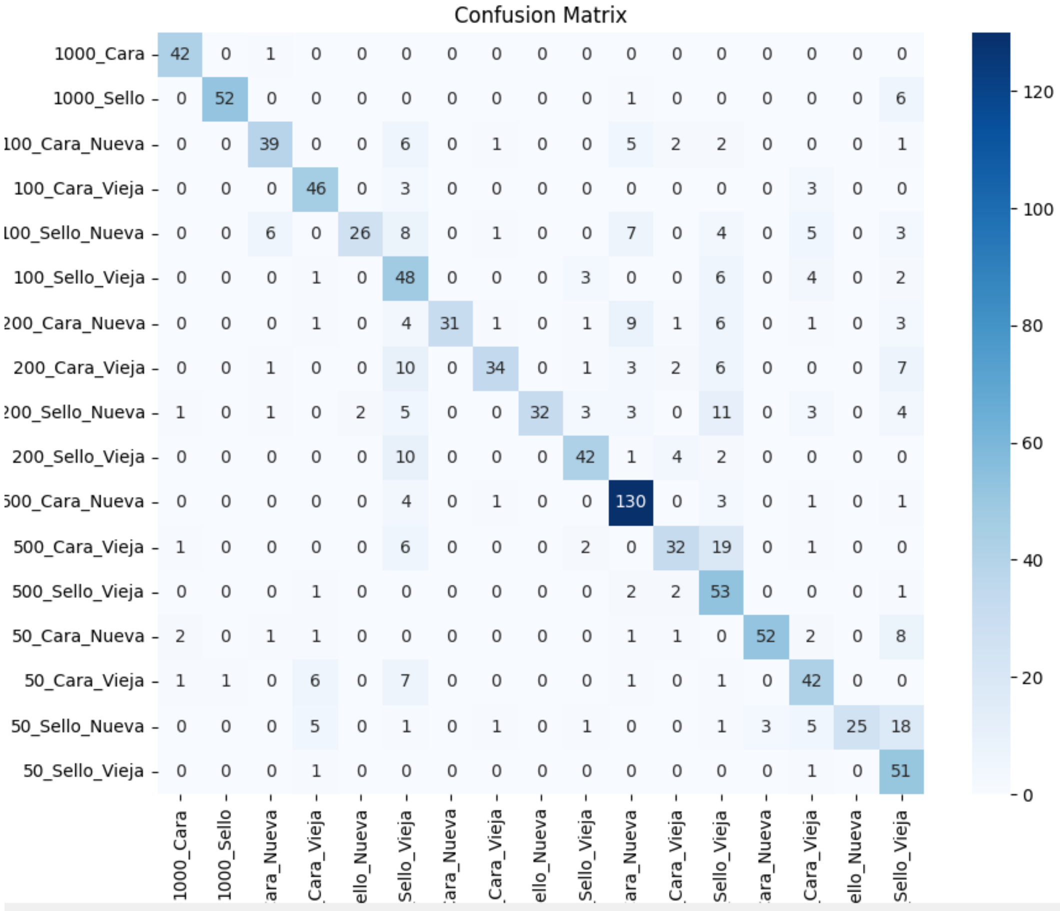
As can be seen, none of the coins has almost any correlation with other coins, i.e., there will be no confusion between them to predict whether or not it is the coin that is being presented in the image.

	precision	recall	f1-score	support
1000_Cara	0.88	0.98	0.92	43
1000_Sello	0.88	0.85	0.86	59
100_Cara_Nueva	0.96	0.96	0.96	56
100_Cara_Vieja	0.83	0.96	0.89	52
00_Sello_Nueva	0.84	0.85	0.84	60
00_Sello_Vieja	0.80	0.88	0.84	64
200_Cara_Nueva	0.90	0.74	0.81	58
200_Cara_Vieja	0.92	0.92	0.92	64
00_Sello_Nueva	0.98	0.92	0.95	65
00_Sello_Vieja	0.87	0.90	0.88	59
500_Cara_Nueva	0.92	0.94	0.93	140
500_Cara_Vieja	0.96	0.80	0.88	61
00_Sello_Vieja	0.90	0.90	0.90	59
50_Cara_Nueva	0.92	0.82	0.87	68
50_Cara_Vieja	0.82	0.86	0.84	59
50_Sello_Nueva	0.85	0.87	0.86	60
50_Sello_Vieja	0.91	0.98	0.95	53
accuracy			0.89	1080
macro avg	0.89	0.89	0.89	1080
weighted avg	0.89	0.89	0.89	1080

The graph below shows a detailed report with metrics that evaluates each class in a classification problem, and this graph is the one that represents the **logistic regression**.

5.2. KNN

For KNN we have the following confusion matrix.



As can be seen, there are some currencies that have a higher correlation with specific currencies. This gives us an indication that there may be coins that become more complex for the model to identify because it confuses them or believes them to be another coin.

	precision	recall	f1-score	support
1000_Cara	0.89	0.98	0.93	43
1000_Sello	0.98	0.88	0.93	59
100_Cara_Nueva	0.80	0.70	0.74	56
100_Cara_Vieja	0.74	0.88	0.81	52
100_Sello_Nueva	0.93	0.43	0.59	60
100_Sello_Vieja	0.43	0.75	0.55	64
200_Cara_Nueva	1.00	0.53	0.70	58
200_Cara_Vieja	0.87	0.53	0.66	64
200_Sello_Nueva	1.00	0.49	0.66	65
200_Sello_Vieja	0.79	0.71	0.75	59
500_Cara_Nueva	0.80	0.93	0.86	140
500_Cara_Vieja	0.73	0.52	0.61	61
500_Sello_Vieja	0.46	0.90	0.61	59
50_Cara_Nueva	0.95	0.76	0.85	68
50_Cara_Vieja	0.62	0.71	0.66	59
50_Sello_Nueva	1.00	0.42	0.59	60
50_Sello_Vieja	0.49	0.96	0.65	53
accuracy			0.72	1080
macro avg	0.79	0.71	0.71	1080
weighted avg	0.79	0.72	0.72	1080

The following graph shows a detailed report with metrics that evaluates each class in a classification problem, and this graph is the one that represents the **KNN**.

6. Results Analysis

6.1. Metrics

Before analyzing the results, we must know the metrics with which we are going to analyze the data.

It is very important to measure how well the model generalizes over unseen data is what defines adaptive versus non-adaptive machine learning models. By using different metrics for performance evaluation, we should be in a position to improve the overall predictive power of our model before we run it for production on previously unseen data [5] [6].

In this case we chose two metrics that we consider appropriate for the selected problem, which are Accuracy, Precision, Completeness and Specificity. The reasons why these metrics were chosen will now be presented.

- Precision: Precision is the proportion of instances classified as positive that are actually positive. It is calculated as TP / (TP + FP), where TP are the true positives and FP are the false positives. In other words, it measures the quality of the model's positive predictions.

- Recall: Recall, also known as sensitivity or true positive rate (TPR), is the proportion of positive instances that were correctly detected by the model. It is calculated as TP / (TP + FN), where TP are the true positives and FN are the false negatives. It measures the ability of the model to find all positive instances.

- -score: The F1-score is the harmonic mean of precision and recall. It is calculated as 2 * (precision * recall) / (precision + recall). The F1-score reaches its best value at 1 (best precision and recall) and its worst value at 0.

- Support: Support is the number of occurrences of each class in the test data.

- Accuracy: Accuracy is the proportion of correct predictions over the total predictions. It is calculated as (TP + TN) / (TP + TN + FP + FN), where TN are the true negatives. It is a measure of the overall accuracy of the model.

- Macro avg: The average of the metrics (precision, recall, f1-score) for all classes regardless of their proportion in the test data.

- Weighted avg: The weighted average of the metrics for all classes, where each metric is weighted by the number of instances of the corresponding class in the test data.

6.2. Logistic regression

As can be seen in the results graph, the accuracy of all the classes is very good (above 80 %), meaning that for this model, it is easier to identify each currency class with the established parameters. As for the recall, there was only one value that fell below 80 %, implying that with that specific class it may be a bit more complicated for the model to identify the positive instances of that currency. But there is a key factor here, as we can see, we have a Support column that indicates the number of times a class appears in the dataset, although it is not the same for each class, the model is able to have an incredible accuracy in predicting which coin appears in the image. The accuracy shows the total effectiveness of the algorithm, being a very good one, exceeding by more than 10 figures the KNN, being 89 %. And the last two metrics talk about the average and weighted average of the precision, recall and f1-score metrics, as they are all at a very high level, these metrics are the same percentage of the accuracy.

6.3. KNN

As can be seen in the results graph, the accuracy of almost all the classes is good (above 70 %), only the accuracy of the classes of some old coins is lower than this percentage, meaning that it will be difficult to confirm these coins for the model. As for the recall there were more complications for the model, there are coins where it is difficult for the model to check if the instance being treated is true or not. But there is a key factor here, as we can see, we have a Support column that indicates the number of times that a class appears in the dataset, so as it is not the same for each one, it means that the model can understand or validate a coin better than others according to its class. The accuracy shows the total effectiveness of the algorithm, being a good one, but not surpassing the logistic regression model. And the last two metrics talk about the average and weighted average of the accuracy, recall and f1-score metrics, clearly, as these are not the best, their averages will be the same.

6.4. Logistic regression with PCA vs KNN?

Logistic regression with PCA was better than KNN for different reasons.

1. **Dimensionality of the data**: PCA reduces the dimensionality of the data by transforming the original characteristics into a smaller set of uncorrelated variables, known as principal components. This can help eliminate noise and redundancy in the data, which makes it easier for the logistic regression model to capture more important patterns and improve its predictive ability.

2. **Sensitivity to scale and distance**: KNN classifies data points based on their proximity to the nearest training points in the original feature space. This means that KNN can be sensitive to the scale of the features and the distance metric used. If the data has a high dimensionality and is not well scaled, this can negatively affect the performance of KNN.

7. Impacts of the solution

This solution may have some impacts on society, either good or bad, so we will present some with respect to the model, and others with respect to the society.

7.0.1. Impacts with respect to the model

- Efficiency and processing time: In addition to accuracy, it is important to consider the efficiency and processing time of the models. Logistic regression with PCA can be faster in terms of training and prediction time compared to KNN, which can be beneficial in real-time applications where fast responses are required.
- Ease of implementation: Logistic regression with PCA can be easier to implement and maintain compared to KNN, especially if a dimensionality reduction approach such as PCA is used to handle the high dimensionality of the images. This can reduce the complexity of the system and make it easier to integrate into existing systems.
- Model interpretability: Logistic regression with PCA can provide greater interpretability compared to KNN, as the regression coefficients can help to understand which features are most important for coin classification. This can be useful in justifying model decisions and gaining the confidence of users and stakeholders.

7.0.2. Impacts with respect to society.

Clearly in this area is where it would benefit the most, since it would be of great help for people not to have to count a large amount of coins that can be received for any purchase, exchange for banknotes, among others.

- Facilitation of commercial transactions: By providing an accurate tool to identify Colombian currencies in images, the solution could facilitate commercial transactions in various sectors of the economy, such as retail, banking and public transportation. This could streamline daily operations and improve overall economic efficiency.
- Fraud and counterfeit prevention: The ability to accurately identify coins could help prevent fraud and counterfeiting, which is especially relevant in a country where currency counterfeiting is a known problem. By detecting and rejecting counterfeit currencies, the integrity of the financial system and citizens' confidence in the national currency can be protected.
- Accessibility for the visually impaired: By integrating coin recognition technologies into accessible applications and devices, the solution could improve the autonomy and quality of life of the visually impaired by facilitating the identification and handling of coins.

8. Conclusions

9. Bibliographic References

Referencias

[1] 4 tipos de tareas de clasificación en el machine learning - elearning actual. Available at https://elearningactual.com/4-tipos-de-tareas-de-clasificacion-en-el-machine-learning/ (2024/03/14).

[2] Hough circles. Available at http://raerpo2.blogspot.com/2011/02/14-hough-circles-en-python.html (2024/03/15).

[3] Comprende principal component analysis — aprende machine learning). Available at https://www.aprendemachinelearning.com/comprende-principal-component-analysis/ (2024/03/16).

[4] Exploratory data analysis (eda) - how to do eda for machine learning problems using python - mlplus. Available at https://www.machinelearningplus.com/machine-learning/exploratory-data-analysis-eda/ (2024/03/17).

[5] Métricas de evaluación de modelos en el aprendizaje automático. Available at https://www.datasource.ai/es/data-science-articles/metricas-de-evaluacion-de-modelos-en-el-aprendizaje-automatico (2024/03/17).

[6] Métricas y puntuación: cuantificar la calidad de las predicciones — documentación de scikit-learn - 0.24.1. Available at https://qu4nt.github.io/sklearn-doc-es/modules/model_evaluation.html#scoring-parameter (2024/03/18).

[7] Fransico Charte. Qué tipos de problemas podemos resolver con técnicas de aprendizaje automático (machine learning). Available at https://www.campusmvp.es/recursos/post/que-tipos-de-problemas-podemos-resolver-con-tecnicas-de-aprendizaje-automatico-machine-learning.aspx (2024/03/14).