Currency Recognition: Computer Vision and Machine Learning

Tiago Filipe Sousa Gonçalves, Vitor Daniel Veigas Minhoto

I. Introduction

COMPUTER Vision is the field of research that aims to transform data from a still or video camera into either a decision or a new representation, taking into account a specific objective. In comparison with human vision, which, at first sight, seems a very simple task (thanks to years of evolution of the human brain), machine vision tasks tend to be more difficult, since computers have to handle signals that are converted into grids of numbers and then extract meaningful information from them [1], [2]. Recently, however, with the increase of computation power and the development of new algorithms, more applications for computer vision have been developed, such as: vision inspection, assembly and material handling, automatic target recognition, photo interpretation, and extraction of three-dimensional structure [3]. On behalf of the *Computer Vision* course, taught by Professor Miguel Coimbra at the Faculty of Sciences of University of Porto, during the 1st Semester of the Academic Year of 2018/2019, we have developed a web application for currency recognition, that uses concepts of feature extraction and machine learning applied to computer vision in order to recognize and classify bank notes.

A. Motivation & Problem

The starting point for the development of this course project was related with the assumption that we can't recognize all the bank notes or coins of all the currencies in the World. With that assumption in mind, we started to realize that, perhaps, computer vision techniques could be useful to deploy an user-friendly application where users could take a picture from any banknote or coin and provide that data as an input and then receive the value and currency as an output. Actually, there are key stakeholders that benefit from these applications, such as banking systems [4].

II. Methodology

As baseline, we based our approach in a paper from Costa *et al* [5], who proposed in 2016 a system capable of recognizing Euro banknotes with high precision. Although the main goal of the course project was to develop an algorithm based in traditional computer vision methods, we also decided to test a different approach based on machine learning algorithms. The algorithms were implemented in Python [6], with the support of libraries such as OpenCV [7], for computer vision techniques like image processing and feature extraction, and scikit-learn [8], for rapid implementation of data processing and machine learning algorithms.

A. Dataset

To perform the experiments we used a dataset [5] that contains Euro banknotes divided into two different sets:

- Train Set, which contains front and back images of 5, 10, 20, 50, 100, 200 and 500 Euro banknotes, with low, medium and high resolutions.
- Test Set, which contains images of the same type of the banknotes present in the other set, except that they are placed in different context scenes.

B. Algorithms and Pipeline

For this approach we propose a pipeline that contains the following steps: image processing, feature extraction, classification, decision output.

1) Image Processing: Regarding image processing, we started by applying a bilateral filter in order to remove most of the possible existent noise while preserving important features such as strong edges [5], [9]. After this noise reduction, to prevent problems related with pictures taken in different light environments, we applied a contrast limited adaptive histogram equalization (CLAHE) [10], to increase image contrast.

2) Feature Extraction: SIFT Algorithm: We decided to use Scale Invariant Feature Transform Algorithm (SIFT) [11] to extract meaningful descriptors to be used as features to our classification and decision tasks. SIFT is based on the assumption that is very difficult to use the same window to detect keypoints with different scale, leading, thus, to the need of using a scale-space filtering approach. In this case, to achieve this, a Laplacian of Gaussian (LoG) is found for the image with different σ values; LoG works as a blob detector that will perform detection in various sizes, related with the values of σ, that will be the scaling parameter. The local maxima across the scale is found and we end up with a list of (x, y, σ) values, which means that there is a potential keypoint (x, y) at σ scale. In fact, due to computational costs, in SIFT, instead of applying a LoG, it is applied Difference of Gaussians (DoG), which is an approximation of LoG. After the computation of DoG, images are searched for local extrema over scale and space. After finding the keypoints localizations, they are refined, through a contrast threshold and edge detection techniques [12], which will eliminate the low-contrast keypoints and the edge keypoints, leaving just the strong interest points. After finding and choosing the keypoints, in order to achieve invariance to image rotation, an orientation assignment has to be performed. This is done by taking the neighbourhood around the keypoint location, based on scale, and the gradient magnitude and direction is calculated in that region. Finally, in order to obtain meaningful descriptors, a neighbourhood is also taken, that takes into account orientation.

3) Feature Extraction: Bag of Visual Words: Although it is possible to use the obtained descriptors as image features, we decided to compute a Bag of Visual Words (BoVW), which is the analogy of Bag of Words (BoW), for image analysis [13], to reduce dimension. This concept is based on the idea that it is possible to create vocabularies of visual words, in this case, visual regions for example, that are specific of a certain image. In practice, to implement this approach, what is done is a clustering on the full descriptor set to obtain the set of visual words that will constitute the dictionary, then use the same clustering model on each individual image descriptor vector and the count the occurrences of each visual word per image, building, thus, an histogram, that will be specific for each image. To perform the clustering of the descriptor, we trained a K-Means algorithm [14] on the entire descriptor set in order to obtain the desired k number of clusters. This algorithm is initialized with k random centroids, that, change, in each iteration, in order to maximize the distance between points that were assigned to different clusters. We then applied the trained model to each individual descriptor set to obtain the counts of each visual word per image and to construct a vector where features correspond to this obtained histogram of visual words. After this approach, it is possible to end up with a feature vector that has k features per image.

4) Classification: Computer Vision Approach: For a traditional *Computer Vision* approach we used feature matching methods, that directly compare descriptors to perform the final decision. To perform this, we decided to use the Fast Library for Approximate Nearest Neighbours (FLANN) [15]–[17], implemented in OpenCV [7]. Actually, FLANN is a group of algorithms based on distance to perform distance measures in high dimensional spaces; the less the distance between two given descriptors sets, the similar the images are. One of the most used algorithms to do this is, for instance, K-Nearest Neighbours [18], where k is the number of neighbours that we want to evaluate, based on the Euclidean Distance [19]. To perform the final decision, one must compare the computed descriptor set, obtained from the test image, with each descriptor set from the training images, and, when the value for the minimum distance is achieved, the corresponding image class is set to the test image.

5) Classification: Machine Learning Approach: For a *Machine Learning* approach, one must bear in mind that, in order to train a classifier, the training algorithm needs to be fed with a feature vector, with dimensions *number of samples x number of features*, and a label vector with the corresponding labels per sample. Also, since we had front and back images per each Euro banknote value, we ended up with 16 text labels, that had to be encoded as number labels, to be used in multiple class classification approach. As a feature vector, we used the BoVW previously obtained. To perform the experiments, we tried three different classifiers:

- Support Vector Machines, which is a state-of-art classifier that learns how to separate different classes based on a hyperplane [20]. It is also sensitive to parameter tuning, since there are several parameters that can be optimized in order to obtain better performances (such as kernel, gamma and margin).
- Logistic Regression, which is a model that passes a linear function through a logistic (also know as sigmoid) function that will output a probability [21]. After this, one can set a decision boundary based on these probabilities to make the decision.
- Multilayer Perceptron, also known as Artificial Neural Network, is a simple neural network model that can also be trained to learn how to classify new samples into classes [22].

To perform the final decision, we first obtain the descriptor set of the test image, then the bag of visual words histogram is computed and we obtain a final feature set that will serve as input tho the trained classifier. The classifier will output a number that corresponds to a specific label, i.e., the image class.

III. Application Development and Deploy

In order to publish the work and making a real world product, a web application was developed and deployed as a way to perform a demonstration of the project. Following the trend of serverless computing [23], the algorithm was uploaded to the cloud and converted to a trigger function that is activated upon the arrival of a request. This request is an image sent through the popular application *Facebook Messenger*. The application, called *Currency Bot*, is presented to the user in the form of a Chat Bot. The Bot receives messages from the user, is capable of distinguishing between different content (text, image, video, and others) and to react accordingly. If an image is sent, the algorithm will be executed in the background, and the Chat Bot will reply with the value of the banknote. If the user sends other type of message, the Chat Bot will ask for an image.

IV. Results Discussion & Conclusions

We were able to develop and deploy a web application that receives images of banknotes and outputs their class (value and currency). In the end, it was the *Computer Vision* approach that performed better and with less computational and time costs. Although we were expecting better results from the *Machine Learning* approach, we think that the abnormal results obtained can be explained by the low number of training samples that were available; it would be interesting, as a future work to acquire more images of Euro banknotes, coins and other currencies as well. Another interesting fact is that we are also capable of getting interesting results without a pre-processing stage, showing that SIFT is indeed a robust algorithm to obtain good image descriptors.

References

[1] E. Trucco and A. Verri, *Introductory techniques for 3-D computer vision*. Prentice Hall Englewood Cliffs, 1998, vol. 201.

[2] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[3] W. E. L. Grimson and J. L. Mundy, "Computer vision applications," *Communications of the ACM*, vol. 37, no. 3, pp. 45–51, 1994.

[4] J. Guo, Y. Zhao, and A. Cai, "A reliable method for paper currency recognition based on lbp," in *Network Infrastructure and Digital Content, 2010 2nd IEEE International Conference on*. IEEE, 2010, pp. 359–363.

[5] C. M. Costa, G. Veiga, and A. Sousa, "Recognition of banknotes in multiple perspectives using selective feature matching and shape analysis," in *Autonomous Robot Systems and Competitions (ICARSC), 2016 International Conference on*. IEEE, 2016, pp. 235–240.

[6] "Python 3.7.1 documentation." [Online]. Available: https://docs.python.org/3/

[7] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[9] S. Paris, P. Kornprobst, J. Tumblin, F. Durand *et al.*, "Bilateral filtering: Theory and applications," *Foundations and Trends® in Computer Graphics and Vision*, vol. 4, no. 1, pp. 1–73, 2009.

[10] S. M. Pizer, R. E. Johnston, J. P. Ericksen, B. C. Yankaskas, and K. E. Muller, "Contrast-limited adaptive histogram equalization: speed and effectiveness," in *Visualization in Biomedical Computing, 1990., Proceedings of the First Conference on*. IEEE, 1990, pp. 337–345.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[12] C. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[13] C.-F. Tsai, "Bag-of-words representation in image annotation: A review," *ISRN Artificial Intelligence*, vol. 2012, 2012.

[14] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 881–892, 2002.

[15] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, 2014.

[16] ——, "Fast matching of binary features," in *Computer and Robot Vision (CRV)*, 2012, pp. 404–410.

[17] ——, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Application VISSAPP'09)*. INSTICC Press, 2009, pp. 331–340.

[18] A. Kataria and M. Singh, "A review of data classification using k-nearest neighbour algorithm," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 6, pp. 354–360, 2013.

[19] J. C. Gower, "Euclidean distance geometry," *The Mathematical Scientist*, vol. 7, pp. 1–14, 1982. [Online]. Available: http://www.convexoptimization.com/TOOLS/Gower2.pdf

[20] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[21] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *Journal of biomedical informatics*, vol. 35, no. 5-6, pp. 352–359, 2002.

[22] M. W. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)a review of applications in the atmospheric sciences," *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[23] I. Baldini, P. Castro, K. Chang, P. Cheng, S. Fink, V. Ishakian, N. Mitchell, V. Muthusamy, R. Rabbah, A. Slominski *et al.*, "Serverless computing: Current trends and open problems," in *Research Advances in Cloud Computing*. Springer, 2017, pp. 1–20.