BoVW-CAM: Visual Explanation from Bag of Visual Words

Arnaldo Vitor Barros da Silva(✉) and Luis Filipe Alves Pereira

Universidade Federal do Agreste de Pernambuco, Avenida Bom Pastor, Garanhuns, Pernambuco 55292-270, Brazil
arnaldovitorbarros@gmail.com

Abstract. Classical computer vision solutions were used to extract image features designed by human experts for encoding visual scenes into vectors. Machine learning algorithms were then applied to model such vector space and assign labels to unseen vectors. Alternatively, such space could be composed of histograms generated using the Bag of Visual Words (BoVW) that compute the number of occurrences of clustered handcrafted features/descriptors in each image. Currently, Deep Learning methods automatically learn image features that maximize the accuracy of classification and object recognition. Still, Deep Learning fails in terms of interpretability. To tackle this issue, methods such as Grad-CAM allow the visualization of regions from input images that support the predictions generated by Convolutional Neural Networks (CNNs), *i.e.* visual explanations. However, there is a lack of similar visualization techniques for handcrafted features. This fact obscures the comparison between modern CNNs and classical methods for image classification. In this work, we present the BoVW-CAM that indicates the most important image regions for each prediction given by the BoVW technique. This way, we show a novel approach to compare the performance of learned and handcrafted features in the image domain.

Keywords: Deep Learning · Class Activation Mapping · Image features

1 Introduction

Deep learning has emerged as a new branch of machine learning. It has proven to be very effective in many computer vision tasks such as image classification [15], object detection [25], image segmentation [8], and others. In addition to reporting high accuracy rates, Deep Learning eliminated the requirement for human experts to design feature extractors since convolutional layers of Convolutional Neural Networks (CNNs) are suited for this task.

However, even in the face of all these advantages, Deep Learning used to fail in interpretability [13]. This attribute may be crucial, especially in high misclassification costs. To attack this black box issue, Zhou *et al.* [24] proposed the Class Activation Map (CAM), which highlights the most significant image

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
J. C. Xavier-Junior and R. A. Rios (Eds.): BRACIS 2022, LNAI 13654, pp. 45–55, 2022.
https://doi.org/10.1007/978-3-031-21689-3_4

regions to produce a prediction by a CNN. This technique modifies the network architecture, replacing the fully connected layers with convolutional layers and a Global Average Pooling (GAP). Then, the channels from the output of the last convolutional layer are weighed by the network parameters that link each element in the GAP output to the neuron of the activated class. As a result, this weighted sum of channels is the final visual explanation provided by CAM. More recently, Selvaraju *et al.* proposed the Grad-CAM [21], this method can be applied to many CNN models without requiring architectural changes. For this, it calculates the gradient of the last convolutional layer concerning the network output, which measures the influence of each cell in the feature map to compose the network prediction.

The huge success of Deep Learning methods currently overshadows classic techniques based on Handcrafted (HC) features for image classification [10]. However, some researchers in the literature suggest a careful comparison between Learned (LN) and HC features. Nanni *et al.* [17] ran an exhaustive comparison between the two approaches in different image domains, from butterfly species classification to cancer detection. Their experiments showed several scenarios where HC features outperformed the LN features in accuracy. In early 2020, Lin *et al.* [12] proposed a random forest to identify Magnetic Resonance (MR) images of livers that are adequate for clinical diagnosis. They reported that HC features outperformed LN features across smaller datasets, *i.e.*, less than 200 images for model training. In 2021, Saba *et al.* [20] investigated the problem of detecting microscopic skin cancer in non-dermoscopic color images. They reported cases where HC features were better than LN features. Finally, in 2022, Silva *et al.* [22] evaluated HC and LN features in the context of violence detection in video frames. Their results showed that LN features can not always be claimed superior since some violent scenes are only detected by HC features.

A widely used image representation technique based on local HC features is the Bag of Visual Words (BoVW) [5]. Concerning the existence of many local descriptors along a single image, a keypoint is referred to as a structure composed of a feature/descriptor vector and an image coordinate to indicate the local region described by such feature/descriptor. The final BoVW image representation is an histogram of the occurrences of clustered handcrafted features/descriptors presented in the given image. Finally, the BoVW histograms may feed a classifier like Support Vector Machine (SVM) [9]. This work proposes a visualization method that allows the interpretation of the most important regions for image classification using BoVW. Several works [12,17,20,22] previously evaluated the accuracy rates obtained by HC and LN features to conclude that they focus on different aspects of the images. However, to the best of our knowledge, such divergence was not demonstrated in the literature at the image domain level.

2 Background

2.1 Keypoints

Keypoints refer to structures for encapsulating the representation of local features along a given image. Therefore, for representing a single image patch, a

keypoint has a feature/descriptor vector that holds information about the image *semantics* locally and a coordinate tuple that localize it within the image. The extraction of keypoints is then composed of at least two main steps for retrieving: *(i)* the keypoint localization, and *(ii)* the keypoint feature/descriptor.

On the one hand, a good keypoint localizer identifies local regions that are potentially distinct along the image. Such uniqueness is crucial for representing the image's elements that allow its identification. Example of algorithms for keypoint localization includes FAST [18], BRISK [11], ORB [19], SURF [3], SIFT [14], and KAZE [2]. On the other hand, a good keypoint descriptor faithfully characterizes image local regions. Example of techniques for extracting keypoint descriptors are BRISK [11], FREAK [1], BRIEF [4], SURF [3], ORB [19], SIFT [14], KAZE [2]. Those are all handcrafted techniques, *i.e.* such algorithms are humanly designed and data invariant.

Keypoint Localization. Keypoint localizers generally try to find more representative image patches in relation to their neighbors. This representation can be through aspects such as corners, colors, or brightness. A classic method for locating keypoints is the Harris Corner Detector. From the dx and dy image gradients, a Harris response map is generated by encoding the magnitude of gray level changes in both horizontal and vertical directions for each 3×3 image window. Finally, each pixel in the image whose Harris response exceeds a predefined threshold τ is assigned as a corner.

Another widely used method is the FAST (Features from Accelerated Segment Test). Considering a Bresenham circle of radius three centered at each pixel in the image, the FAST compares the gray value of the central pixel to each intensity along the Bresenham circumference. If an amount of N consecutive pixels of this circumference is brighter or darker than the central point, it is classified as a corner. To speed up the method, it is possible to use a machine learning-based approach for detecting consecutive patterns in a sequence. Then, after extracting these 16-pixel circumferences and their central intensity values, it is possible to train a classifier as a decision tree [16] to decide whether or not this point is a corner.

Other methods like SIFT [14], SURF [3] and KAZE [2] uses multiscale analysis. SIFT algorithm, for instance, computes the Difference of Gaussians (DoG) between different image scales. The local minima and maxima along the DoG are considered keypoint candidates.

Keypoint Descriptors. After the keypoint localization step, it is necessary to associate them with appropriate feature/descriptor vectors that correctly encode their semantics. Such generated descriptors are usually based on histograms of gradients, directions of border orientations, or pixel intensities. For example, using the pixel intensity, we have the BRIEF [4] and FREAK [1] that build the feature/descriptor vectors from the relative intensity of pairs of pixels within the keypoint neighborhood.

Descriptors based on gradients have been more used once they present greater efficiency with lighting variation, resizing, and orientation [2]. In SIFT, for instance, the vectors are constructed within 16 subareas around the keypoint. For each subarea, a histogram of the gradient flow is computed along eight directions. Then, by concatenating the histograms of each subarea, a final feature/descriptor of 128 dimensions is created.

2.2 Bag of Visual Words (BoVW)

The main idea of BoVW is to create new representations of images as histograms. These histograms are relative to the number of occurrences of specific features/descriptor referred to as visual words. To build these histograms, the following steps are necessary: *i)* the features/descriptors of a subset of the data are grouped using some clustering algorithm like K-Means [7], the centroids $\Omega = \{\omega_1, \omega_2, \cdots, \omega_n\}$ resulting from this grouping are then called visual words; *ii)* given the visual dictionary Ω, all features/descriptors extracted within a new image are associated with the visual word closest to them; *iii)* finally, the histogram that will describe this image is generated by computing the number of occurrences of each word in the image. These steps are summarized in Fig. 1.

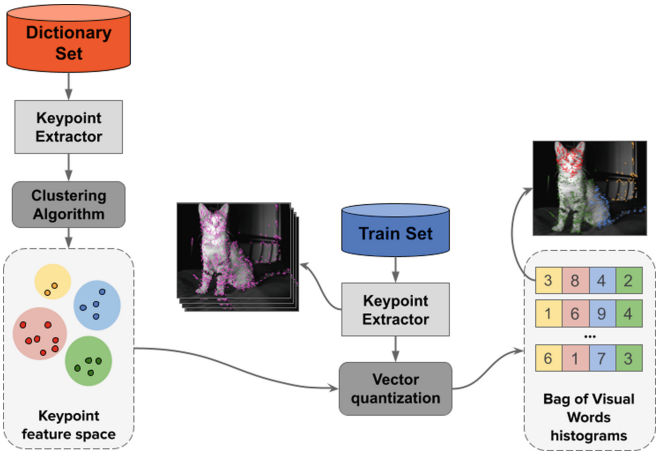

Fig. 1. The Bag of Visual Words (BoVW) working diagram. From a dataset partition (referred to as Dictionary Set), keypoints are localized within all images and their feature/descriptors are extracted. A new vector space of feature/descriptors is then created. By grouping the feature/descriptors using a clustering algorithm, a set of *visual words* $\Omega = \{\omega_1, \omega_2, \cdots, \omega_n\}$ is created in the keypoint feature/descriptors space. Given a new image $\mathbf{x}$ from the Train Set partition, image keypoints are localized and their feature/descriptors are extracted. Finally, in a vector quantization step, a frequency histogram compute how many keypoints of $\mathbf{x}$ falls into each word of Ω.

3 Methodology

The proposed Class Activation Mapping (CAM) technique for visualizing significant regions of the image that support the current BoVW prediciton can be divided into three steps: *(i)* generating a correlation matrix between words $\omega_k, 1 \leq k \leq K$ (for K visual words) and labels $c_j, 1 \leq j \leq J$ (for J classes), *(ii)* generating a visual heatmap for highlighting the words along the image domain, and *(iii)* post-processing the BoVW-CAM visualization. These steps are graphically represented in Fig. 2.

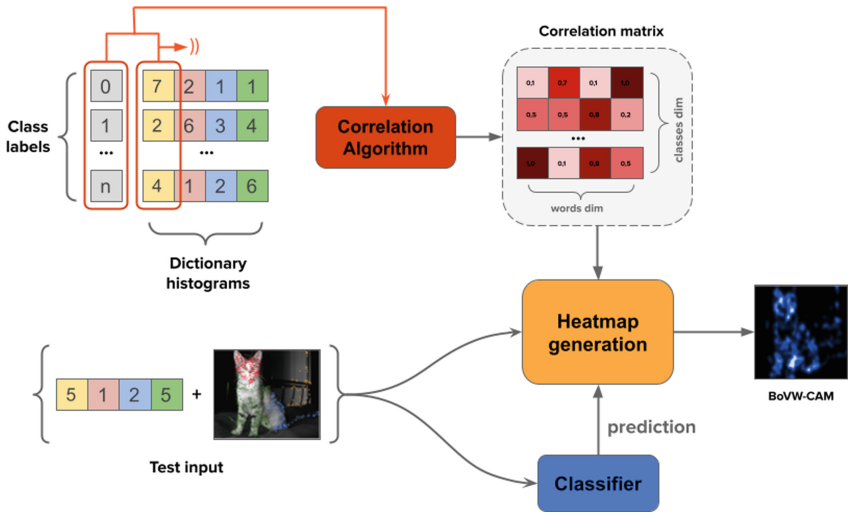

Fig. 2. The BoVW-CAM working diagram. The correlation between each visual word $\omega_k, 1 \leq k \leq K$ (for K visual words) from the dictionary Ω and the classes $c_j, 1 \leq j \leq J$ (for J classes) in the dataset are calculated to generate a $J \times K$ correlation matrix. Finally, given a new test input composed by image, keypoint, BoVW histogram, and class predicted, each keypoint location in the image domain is highlighted according the correlation of its closest visual word and the predicted class to generate a visual explanation accordingly to the BoVW-CAM.

In the first step, using the feature/descriptors ω_k that compose the dictionary Ω of visual words, correlation coefficients between the visual words $\omega_k, 1 \leq k \leq K$ and each problem class $c_j, 1 \leq j \leq J$ are calculated using the Spearman's rank correlation coefficient algorithm [23]. Therefore, a correlation matrix is generated where each column represents a dictionary word ω_k, and each line represents a classification label c_j.

In the second step, an image heatmap is generated from *(a)* an input image, *(b)* its BoVW histogram, *(c)* its keypoints, and *(d)* the predicted label. Then, each keypoint location in the image domain is highlighted according the correlation of its closest visual word and the predicted class to generate a visualization of the most important keypoints.

Finally, in the third step, operations are applied to improve the previous visualization as a heatmap (Fig. 3). First, a MaxPooling2D is used to facilitate the visual identification of image regions densely occupied by keypoints, followed by Gaussian Blur to attenuate the gray value variations to induce a smooth heatmap. Since the MaxPooling2D is an operation that reduces the input dimension, upsampling the image back to the initial size is necessary. Then, we then have the final BoVW-CAM view relative to the target class. The whole method can be seen in details in the Algorithm 1.

Algorithm 1: The BoVW-CAM method

Input: $dict\_hists, class\_labels, kp\_list\_test, img\_test, pred\_test$
Output: $feature\_map$
$corr\_matrix \leftarrow []$
for *each* $label \in class\_labels$ **do**
  $line \leftarrow []$
  for *each* $column \in dict\_hists$ **do**
    $line.add(corr(column, label))$
  end
  $corr\_matrix.add(line)$
end
$feature\_map \leftarrow zeros(img\_test.width, img\_test.height)$
for *each* $kp \in kp\_list\_test$ **do**
  $feature\_map[kp.X][kp.Y] \leftarrow corr\_matrix[pred\_test][kp.Cluster]$
end
$feature\_map \leftarrow max\_pooling(feature\_map)$
$feature\_map \leftarrow gaussian\_blur(feature\_map)$
$feature\_map \leftarrow resize(feature\_map, img\_test.width, img\_test.height)$

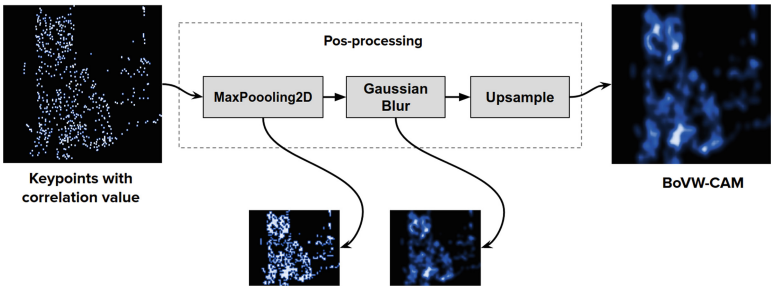

Fig. 3. Scheme for post-processing the visualization of the most important keypoints for generating thee final BoVW-CAM heatmap. The input image goes through a MaxPooling2D to facilitate the visual identification of image regions densely occupied by keypoints, after that a Gaussian Blur is used to smooth the image gray values to create a smooth heatmap, and finally the image is upsampled to its original size.

4 Experiments

We designed experiments for comparing the most important image regions for classification via Bag of Visual Words and Convolutional Neural Networks (CNNs). To the best of our knowledge, this visual comparison in the image domain is unprecedented in the state-of-the-art.

For the experiments, we used the "Cats vs. Dogs"[1] dataset, which is a standard benchmark for binary image classification. In total, the set is composed of 12,500 images for each class.

4.1 Experimental Parameters

We used SIFT as keypoint extractor, the classifier used with the BoVW was the SVM, and the clustering algorithm was the K-Means. Finally, we used 256 words to construct the dictionary. With respect to the CNN architecture, we used three convolutional layers followed by two fully connected layers. The ReLU activation function was employed in all the layers except for the last one which was activated by Sigmoid. The evaluated architecture can be seen in Fig. 4. The optimization technique was the RMSProp, the loss function was Binary Crossentropy, the learning rate was 0.001, and the training lasted for 20 epochs.

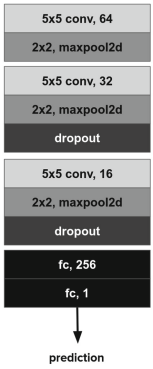

Fig. 4. CNN architecture used in this work.

The database was divided following two distinct approaches: for training the CNN, 70% of the data were used. For training the BoVW, the previous training partition was divided into two folds of the same size, one for building the dictionary and another for training the classifier. The other partitions in both approaches were made in the same proportion, 10% for validation and 20% for testing.

[1] https://www.kaggle.com/datasets/shaunthesheep/microsoft-catsvsdogs-dataset.

5 Results

5.1 Visualization

We generated visual explanations for classifications accordingly the learned features via Grad-CAM [21] and the handcrafted features via the proposed BoVW-CAM in Figs. 5 and 6. It is clear that the two approaches focus on different aspects of the images; the BoVW method seems to cover a larger area of the classified object, while CNN focuses on fewer aspects of the image.

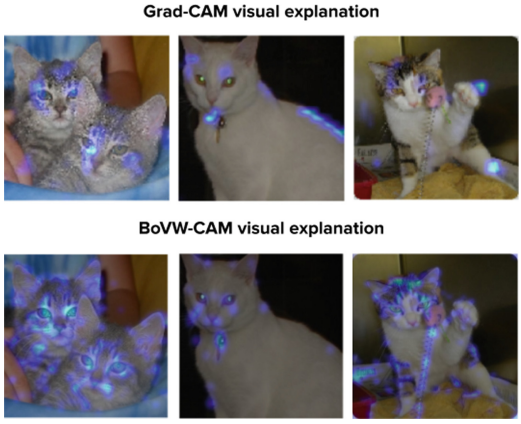

Fig. 5. Visualization for cat class with Grad-CAM and BoVW-CAM methods.

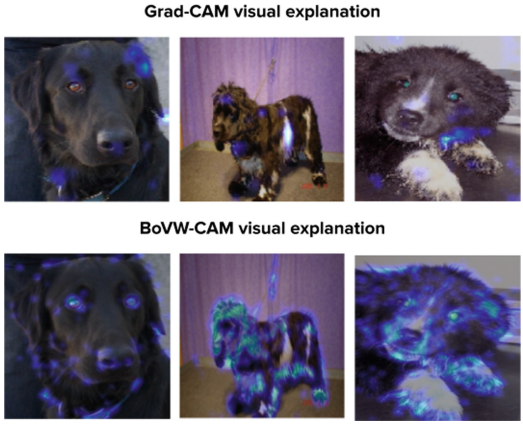

Fig. 6. Visualization for dog class with Grad-CAM and BoVW-CAM methods.

5.2 Venn Diagram

It is also possible to reinforce the hypothesis that learned and handcrafted features are focused on different aspects of the evaluated images by building a Venn Diagram of their predictions. In Fig. 7 we can see that 66.45% of the test set are corrected classified by both methods and the CNN classifies correctly more than BoVW. However, a significant amount of images (523) are misclassified by the CNN while corrected classified by the BoVW. This strengthens the fact that it is not so straightforward that Deep Learning methods can totally replace classical methods based on handcrafted features.

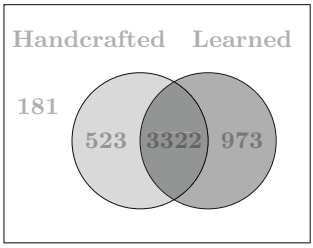

Fig. 7. Number of samples corrected classified by the handcrafted and learned features.

5.3 Dice Score

To measure how big is the difference in the image focus between BoVW and CNN, we transformed the visual explanations of Grad-CAM and BoVW-CAM into binary images for the entire test set for calculating the Dice Score [6] between them. As a result, an average of 0.359 with a standard deviation of 0.138 was obtained. This result confirms that there is a high divergence between the aspects observed by handcrafted and learned features.

6 Conclusion

Based on classification accuracy rates, previous works have suggested that there is a divergence between the aspects that handcrafted and learned features focus on images. In this work, we developed a method capable of generating visual explanations for classification algorithms based on BoVW. Then, we could compare our results with the visual explanations generated by a Grad-CAM on a CNN. In this work, we visually compared the most relevant image regions for classifications based on handcrafted features based on keypoints and learned features. The quantitative evaluation via DICE score confirms that the pixels considered by each classification method highly diverge from each other. Furthermore, despite the Deep Learning method having achieved a higher accuracy rate, we showed a significant amount of test data corrected classified exclusively by the BoVW.

References

1. Alahi, A., Ortiz, R., Vandergheynst, P.: Freak: fast retina keypoint. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 510–517. IEEE (2012)
2. Alcantarilla, P.F., Bartoli, A., Davison, A.J.: KAZE features. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7577, pp. 214–227. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33783-3_16
3. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006). https://doi.org/10.1007/11744023_32
4. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: binary robust independent elementary features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_56
5. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV, Prague, vol. 1, pp. 1–2 (2004)
6. Dice, L.R.: Measures of the amount of ecologic association between species. Ecology **26**(3), 297–302 (1945)
7. Forgy, E.W.: Cluster analysis of multivariate data: efficiency versus interpretability of classifications. Biometrics **21**, 768–769 (1965)
8. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Garcia-Rodriguez, J.: A review on deep learning techniques applied to semantic segmentation. arXiv preprint arXiv:1704.06857 (2017)
9. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intell. Syst. App. **13**(4), 18–28 (1998)
10. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
11. Leutenegger, S., Chli, M., Siegwart, R.Y.: Brisk: Binary robust invariant scalable keypoints. In: 2011 International Conference on Computer Vision, pp. 2548–2555. IEEE (2011)
12. Lin, W., Hasenstab, K., Moura Cunha, G., Schwartzman, A.: Comparison of handcrafted features and convolutional neural networks for liver MR image adequacy assessment. Sci. Rep. **10**(1), 1–11 (2020)
13. Lipton, Z.C.: The mythos of model interpretability: in machine learning, the concept of interpretability is both important and slippery. Queue **16**(3), 31–57 (2018)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (2004)
15. Lu, D., Weng, Q.: A survey of image classification methods and techniques for improving classification performance. Int. J. Remote Sens. **28**(5), 823–870 (2007)
16. Myles, A.J., Feudale, R.N., Liu, Y., Woody, N.A., Brown, S.D.: An introduction to decision tree modeling. J. Chemom. J. Chemom. Soc. **18**(6), 275–285 (2004)
17. Nanni, L., Ghidoni, S., Brahnam, S.: Handcrafted vs non-handcrafted features for computer vision classification. Pattern Recogn. **71**, 158–172 (2017)
18. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006). https://doi.org/10.1007/11744023_34

19. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to sift or surf. In: 2011 International Conference on Computer Vision, pp. 2564–2571. IEEE (2011)
20. Saba, T.: Computer vision for microscopic skin cancer diagnosis using handcrafted and non-handcrafted features. Microsc. Res. Tech. **84**(6), 1272–1283 (2021)
21. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626 (2017)
22. da Silva, A.V.B., Pereira, L.F.A.: Handcrafted vs. learned features for automatically detecting violence in surveillance footage. In: Anais do XLIX Seminário Integrado de Software e Hardware, pp. 82–91. SBC (2022)
23. Zar, J.H.: Spearman rank correlation. Encyclop. Biostatist. **7**, 1–7 (2005)
24. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929 (2016)
25. Zou, Z., Shi, Z., Guo, Y., Ye, J.: Object detection in 20 years: a survey. arXiv preprint arXiv:1905.05055 (2019)