Image clustering - Bee & Wasps

Suto Robert-Lucian (411)

1 Introduction

The following documentation presents the outcome of various unsupervised tasks applied to a selected dataset through various feature extraction methods. The two specific feature extraction methods and the results of applying three clustering algorithms (AgglomerativeClustering, KMeans, and MiniBatchKMeans) to the dataset will be presented and compared to both random chance and a supervised baseline.

2 The Dataset

The dataset that was used for this project can be found at the following link, and contains a collection of 8126 images, depicting bees and wasps. Both classes are split into 2 folders, one containing clear images of the class, and the second one which contains blurry pictures, and often, taken in the same place. Considering this, I decided to drop both of the folders with blurred pictures, ending up with 2127 pictures of wasps and 2469 pictures of bees, therefore we could consider the dataset balanced and labeled.

Most of the pictures contain either a bee or a wasp, or them on various plants, but there are some examples with multiple bees or wasps. Before proceeding with preprocessing the dataset and building the models, I divided the data into 80% for training and 20% for testing in order to evaluate the performance of the clustering algorithms discussed later in this documentation. Considering the small size of the subjects in some photos, the images are resized to 120 x 120.

Figure 1: Examples of images found in the dataset

3 Feature Extraction

An important aspect of this project is feature extraction, which aims to identify important information from the original images[5]. I examined two feature extraction methods and applied them to the algorithms, using the two selected types of features. The methods will be discussed in further detail below.

3.1 Color Spaces

This approach method consists in converting the images to different color spaces, then computing the mean of the n-channels for all of the pixels in an image, reducing the dimensions, while still having relevant features. Some of the color spaces used were Grayscale, CMYK and HSV. Out of these three color spaces, I have decided to go with the HSV one, since it gave slightly better results.

3.2 Bag of Visual Words

For the second method, I chose to extract feature vectors into a bag of visual words (BOVW). Using a SIFT descriptors, I've extracted a collection of descriptors, or, in short, a group of patches, describing some of the more important features of an image[4].

To generate a visual dictionary, I utilized a KMeans clustering algorithm with 80 clusters, placing each element in the image to the nearest centroid. After that, I computed a histogram for each image based on the clusters assigned to the image's extracted patches and employed these histograms as the training and testing features. (The number of bins was usually around 1000 for each histogram)

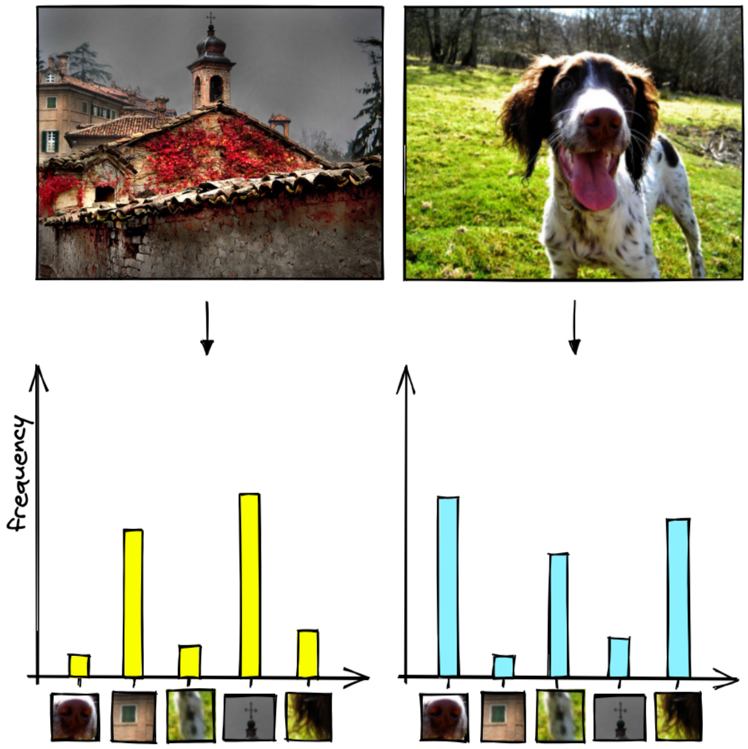


Figure 2: Example of how BOVW extracts and classifies histograms

4 Clustering Algorithms

After extracting features from the images, I will now present how well the three different clustering algorithms perform, using each type of features and comparing their results, using different values for the hyper-parameters.

4.1 Agglomerative clustering

Agglomerative clustering is a bottom-up approach to clustering where each data point is initially considered as its own cluster. The algorithm then merges the most similar ones into a single "agglomeration"[1]. The results are as follows:

Features	linkage	affinity	Train acc	Validation acc	Silhouette Score
BOVW	average	euclidian	0.537	0.534	0.876
BOVW	average	manhattan	0.537	0.468	0.833
BOVW	ward	euclidian	0.420	0.532	0.213
BOVW	complete	euclidian	0.536	0.484	0.865
BOVW	complete	manhattan	0.499	0.497	0.857
HSV	ward	euclidian	0.443	0.454	0.683
HSV	average	euclidian	0.442	0.453	0.821
HSV	complete	euclidian	0.536	0.484	0.876

As we can see, usually, the BOVW feature extraction works the best, but, it's not necessarily a good clasification method, giving us results slightly above the random chance. Computing the confusion matrix on the train data, we can observe that the model, is having a hard time placing the images in clusters. The first label which represent the bees, has assigned 1611 image, out of which only 801 are correctly assigned, the rest of 810 being wasps, and the second one, which is composed of 1606 images, only 840 images are correctly placed .

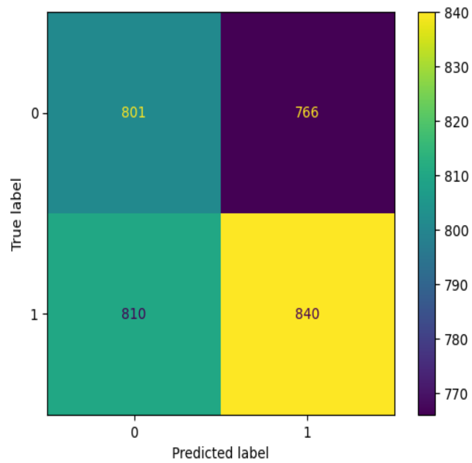
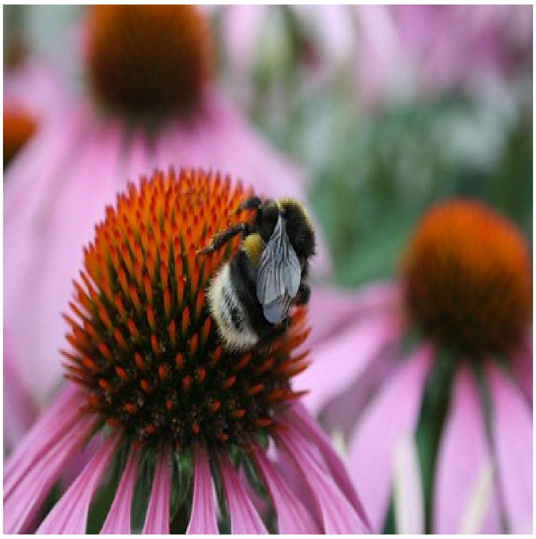


Figure 3: Example of how BOVW extracts and classifies histograms

(a)



(b)

Figure 4: Example of mispredicted images by AgglomerativeClustering

4.2 KMeans

The KMeans algorithm groups the training images into clusters, with the goal of predicting which cluster unseen images belong to[2]. Since the dataset is labeled with two classes, the algorithm is able to group the data into two clusters. The pre-existing knowledge of the number of classes (2) is used to set the number of clusters in the KMeans model. The results are as follows:

Features	init	algorithm	tol	Train acc	Validation acc	Silhouette Score
HSV	k-means++	elkan	1e-4	0.523	0.548	0.177
HSV	k-means++	elkan	1e-2	0.643	0.621	0.241
HSV	k-means++	full	1e-4	0.646	0.625	0.240
HSV	k-means++	full	1e-2	0.644	0.621	0.241
BOVW	k-means++	elkan	1e-4	0.605	0.584	0.229
BOVW	k-means++	full	1e-4	0.561	0.545	0.237
BOVW	k-means++	elkan	1e-2	0.585	0.588	0.210
BOVW	k-means++	full	1e-2	0.542	0.546	0.221

(a) (b)

Figure 5: Example of mispredicted images by MiniBatchKMeans

4.3 MiniBatchKMeans

I've also selected the MiniBatchKMeans algorithm as a bonus, which is a variation of KMeans that utilizes batches of images stored in memory to update the clusters at each iteration[3]. The outcome of this clustering algorithm is:

Features	batch_size	Train acc	Validation acc	Silhouette Score
HSV	200	0.607	0.606	0.262
HSV	500	0.644	0.62	0.242
HSV	1000	0.627	0.618	0.252
HSV	2000	0.625	0.625	0.239
BOVW	200	0.572	0.584	0.221
BOVW	500	0.575	0.589	0.231
BOVW	1000	0.599	0.590	0.239
BOVW	2000	0.593	0.602	0.244

(a) (b)

Figure 6: Example of mispredicted images by KMeans

5 Comparison

As we can see from the reults, the performance of the clustering algorithms is not optimal. However, according to the accuracy scores, the KMeans and MiniBatchKMeans models, applied to features extracted using BOVW, perform the best, but don't have the best silhouette scores. Almost every algorithm is similar to the random chance, the best ones beating it with only 10%.

5.1 Random Chance

The Random chance was computed by assigning random values of 0,1 for the output labels values and comparing it to the true labels. Considering the fact that we only have two classes, the random chance is almost 50% 50% everytime, rarely going bellow 46.5%.

5.2 Supervised Baseline

To establish a supervised baseline, I trained a SVM model (using default hyperparameter values) on the training set after extracting features from the images. Using the first extracting method, the SVM achieved a 82% accuracy on the train set, and 72% on the test one. For the second method, it achieved 77% on the train set, and 70% on the test one.

In conclusion, the results of the experiment show that the supervised SVM model performed better than the unsupervised clustering algorithms on this dataset. The SVM models achieved higher accuracy scores using both feature extraction methods and even using minimal preprocessing.

References

[1] sklearn.cluster.AgglomerativeClustering Documentation. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html. [Accessed 22.01.2023].

[2] sklearn.cluster.KMeans Documentation. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. [Accessed 22.01.2023].

[3] sklearn.cluster.MiniBatchKMeans Documentation. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MiniBatchKMeans.html. [Accessed 22.01.2023].

[4] Bethea Davida. Bag of Visual Words in a Nutshell. https://towardsdatascience.com/bag-of-visual-words-in-a-nutshell-9ceea97ce0fb, 2018. [Accessed 23.01.2023].

[5] Saurabh Srivastava A. K. Sinha Rajkumar Goel, Vineet Kumar. A Review of Feature Extraction Techniques for Image Analysis. https://ijarcce.com/upload/2017/si/ICACTRP-17/IJARCCE-ICACTRP%2029.pdf, 2017. [Accessed 24.01.2023].