University of Bucharest

Faculty of Mathematics and Informatics

2025–2026

Practical Machine Learning
Unsupervised Horse vs Zebra separation

Popescu Pavel–Yanis

Group 407 — Artificial Intelligence

Contents

1 Dataset

1.1 Horse2zebra dataset

The dataset selected for this project is Horse2zebra[1], an image dataset containing horses and zebras. It consists of 1187 horse images and 1474 zebra ones, organized into training and test subsets for each of the two animal domains. Each image has an associated domain label which is used after clustering, mainly for computing comparison metrics.

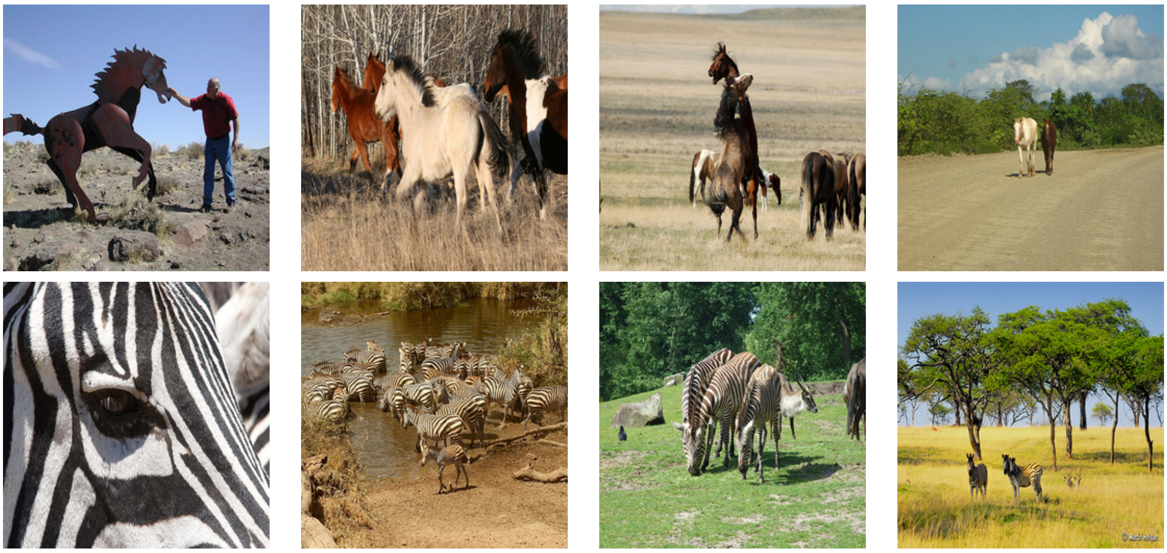

Figure 1: Sample images from the Horse2Zebra dataset. The first row contains horse images, while the second row contains zebra images.

1.2 Train/test split and data handling

As stated above, the dataset is split into a training and a test set. The training set is used for feature extraction, unsupervised model fitting, hyperparameter tuning using unsupervised criteria.

The test split, on the other hand, is used only for the final evaluation after the best model is selected, ensuring that there is no leakage of test data into the model selection process and ensuring the correctness of the approach.

2 Feature representations

The project evaluates two different handcrafted types of feature representations, **Histogram of Oriented Gradients** and **Local Binary Patterns**.

2.1 Histogram of Oriented Gradients (HOG)

The Histogram of Oriented Gradients is a type of feature descriptor that focuses on the shape of the objects by aggregating the directions of local edges in an image. It basically encodes information about object boundaries and rough structural patterns, such as legs, torso outlines, and pronounced intensity changes.

In my implementation, HOG is computed on the grayscale version of the center-cropped image (keeping the central 70% of the original 256×256 image) in order to reduce background noise and focus on the animal region (usually in the center). The HOG vectors that result are standardized by statistics computed on the training set, and then reduced using PCA to 10 components for efficiency.

For the Horse2Zebra dataset, HOG features are relevant, because zebras usually have strong and repetitive edge patterns due to their stripes, while horses tend to produce smoother gradients and more uniform regions.

2.2 Local Binary Patterns (LBP)

Local Binary Patterns is a texture-based descriptor that encodes local intensity contrasts between a pixel and its neighbors. The resulting LBP codes are combined into a histogram, which captures detailed texture patterns.

In my implementation, LBP is computed on the grayscale version of the center-cropped image (keeping the central 70% of the original image). I use uniform LBP with $P = 8$ neighbors and radius $R = 1$, that create a 10-bin normalized histogram representation for each of the images.

For the Horse2Zebra dataset, LBP is good, because zebra stripes create local texture transitions that produce specific LBP patterns, while horse images tend to have smoother and less repetitive textures.

3 Unsupervised learning methods

As unsupervised learning methods, two clustering approaches were chosen: **Fuzzy C-Means** and **Agglomerative Hierarchical Clustering**. Both of them are trained on the same feature spaces (HOG and LBP), creating, in this way, basically four experimental pipelines.

3.1 Fuzzy C-Means (FCM)

Fuzzy C-Means is a soft clustering method, which computes a membership vector for each sample, that describes how strongly that sample belongs to each computed cluster, instead of assigning each sample to just one cluster, like k-means. FCM minimizes a weighted within-cluster objective, which is controlled by a fuzziness parameter m. Higher values for m produce softer memberships, while values closer to 1 approach hard clustering, at 1 becoming basically k-means.

In my project, FCM is applied with two clusters, since we only have zebras and horses as the animal domains. The model is trained only on the training split, using the two HOG and LBP feature representations extracted previously. I try multiple values for the fuzziness parameter m and also different types of distance metrics (Euclidean, Manhattan and Cosine).

The way the hyperparameters are tuned is based on the Fuzzy Partition Coefficient (FPC), which is also computed just on the training data and the combination of hyperparameters that maximizes it is selected as being the best one.

After the training part, the cluster memberships are converted to hard assignments by selecting the maximum membership value per sample. The cluster labels are mapped to labels representing the two animal classes by using a majority-vote approach on the training set. The resulting mapping (0 -> Zebra or 0 -> Horse and vice versa) is then applied to the test set to evaluate the predictions.

3.2 Agglomerative Hierarchical Clustering (AHC)

Agglomerative hierarchical clustering is a bottom-up approach that initially considers every sample as its own cluster and then merges them one by one based on a linkage rule and some chosen distance metric. To get the final clustering the hierarchy is cut at two clusters, representing the horse and zebra domains in our case.

In the project, AHC is only used on the train data subset, using the two HOG and LBP feature representations, for which I evaluate hyperparameter combinations like Ward linkage with Euclidean distance and Average, Single and Complete linkage with Cosine distance.

Agglomerative clustering does not have its own prediction mechanism for testing data, so I compute the clusters' centroids and then assign the test data's cluster by selecting the closest centroid based on the same distance metric chosen in training.

Hyperparameter tuning is done by maximizing the silhouette score, that is computed just from the training subset. The linkage-metric combination that produces the highest one is selected as the desired configuration.

After the clustering process, the cluster labels are mapped to the horse and zebra classes with a majority-vote strategy on the training set. Then that mapping is applied to the test set so the final predictions can get evaluated.

4 Experimental protocol

4.1 Train/test separation and prediction protocol

Both of the clustering methods are trained only on the train set. In the case of Fuzzy C-Means, cluster memberships for the test set are obtained by applying the fuzzy inference step using the centroids learned during training, keeping the same fuzziness parameter and distance metric.

For the Agglomerative approach we do not have a native prediction mechanism, so the cluster centroids are computed from the training data after clustering and the test samples are assigned to those clusters based on the closest centroid from that point, according to the same distance metric used in training. The test data is never used in the hyperparameter tuning process, nor for centroid computation.

4.2 Consistency across experimental pipelines

The four experiments differ just in the feature representation used (HOG vs LBP) and the clustering method (FCM vs AHC), making the comparison between the configurations easy to do.

The dataset train/test split, preprocessing and the evaluation steps are the same for all the experiments so that any performance difference can be interpreted as because of the clustering algorithm or feature representation type.

5 Hyperparameter tuning

In this section, the main focus will be the unsupervised hyperparameters selection process and the actual quality of the clusters.

5.1 Fuzzy C-Means hyperparameter selection

For Fuzzy C-Means, the fuzziness parameter m is chosen by selecting the one which maximizes the Fuzzy Partition Coefficient, only evaluated on the train subset.

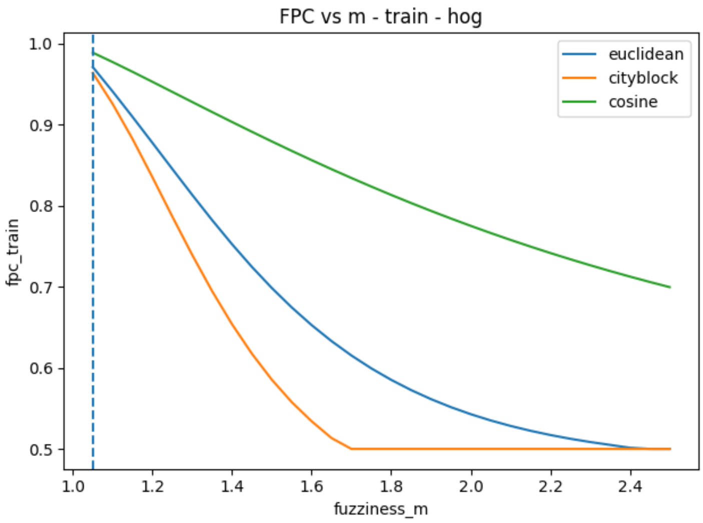

Figure 2: FCM hyperparameter tuning for HOG features: maximizing FPC for m (train only).

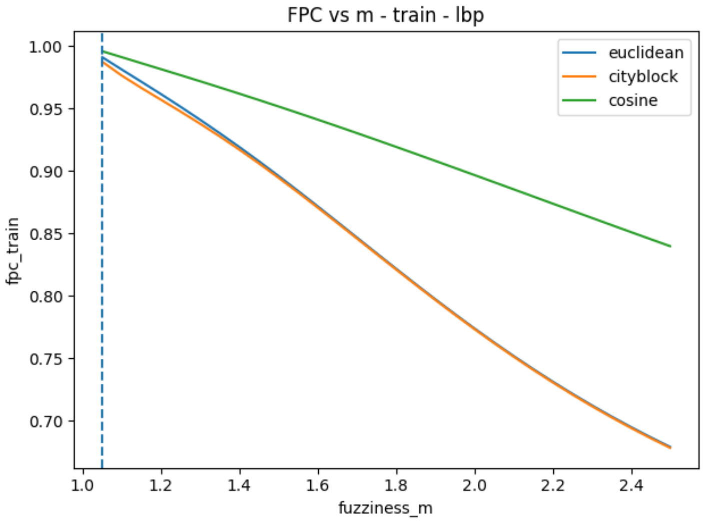

Figure 3: FCM hyperparameter tuning for LBP features: maximizing FPC for m (train only).

It is easily visible that for both of the HOG and LBP features, the Fuzzy Partition Coefficient's value is higher for values of m closer to 1, meaning that increasing m does not improve the quality of the partition, but it really leads us to having more diffuse memberships.

This happens because in our feature spaces, the data already supports clusters that are relatively well-defined and, therefore, Fuzzy C-means tends to favor those solutions that are closer to hard assignments, almost like the behavior of k-means ($m \to 1$) so in our case, a higher m introduces ambiguity rather than capturing any meaningful overlap between the two clusters.

5.2 Agglomerative clustering hyperparameter selection

For Agglomerative Hierarchical Clustering, combinations of linkage strategies and distance metrics are evaluated using the silhouette score on the train subset, the winner being the combination with the highest one.

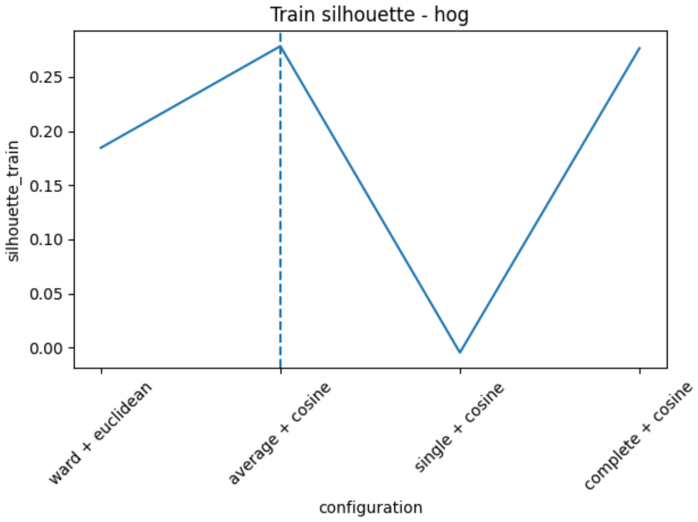

Figure 4: AHC hyperparameter tuning for HOG features: maximizing silhouette for linkage and distance combinations (train only).

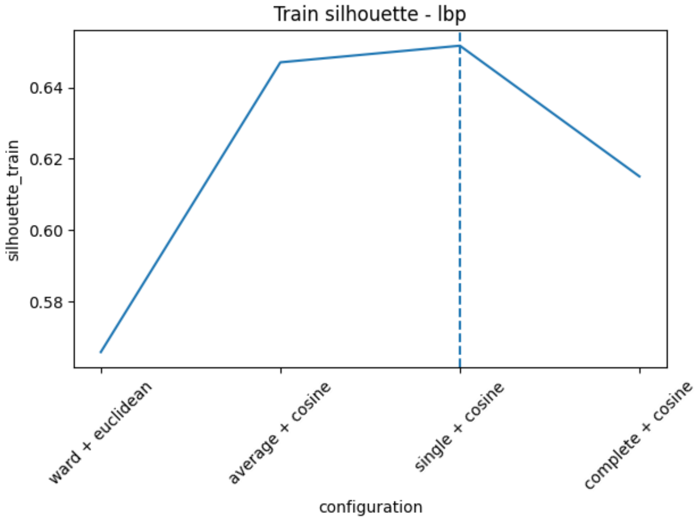

Figure 5: AHC hyperparameter tuning for LBP features: maximizing silhouette for linkage and distance combinations (train only).

As we can see, only 4 linkage-distance combinations are evaluated, based on theoretical constraints and our feature representation space. Ward linkage with Euclidean distance minimizes the within-cluster variance. On the other hand, cosine distance made sense because both HOG (after dimensionality reduction with PCA) and LBP, represent images in a high-dimensional space and angular similarity is usually more informative than the absolute magnitude, so, for cosine distance metric, the single (minimum distance between clusters), complete (maximum distance between clusters) and average (average distance between clusters) linkage were evaluated.

6 Baselines: random chance and supervised comparison

6.1 Random baseline

The random baseline quantifies performance relative to chance and in our case, since the problem is one of binary classification, a uniform random classifier has an accuracy close to 50%. This random baseline is also averaged over 1000 independent trials and both mean accuracy and standard deviation are reported.

Trials	Chance	Train Acc. (mean)	Train Acc. (std)	Test Acc. (mean)	Test Acc. (std)
1000	0.50	0.4995	0.0100	0.5000	0.0312

Table 1: Random baseline performance averaged over 1000 trials

6.2 Supervised KNN baseline

For the supervised baseline, I chose a KNN classifier on the same two feature representations (HOG & LBP), the goal being not to improve the score of the unsupervised models, but rather to estimate how much better a model can separate the two domains when it has access to labels during training.

Of course, in order to avoid any leakage, from the train split, I extract some stratified validation split (80% train, 20% validation). Then a grid search is performed over k number of neighbors (odd numbers from 1 to 499), with different neighbor weights (`uniform` vs. `distance`) and distance metric (`minkowski`, `euclidean`, `manhattan`, `cosine`) hyperparameter combinations, choosing the best one as the one with the highest validation accuracy.

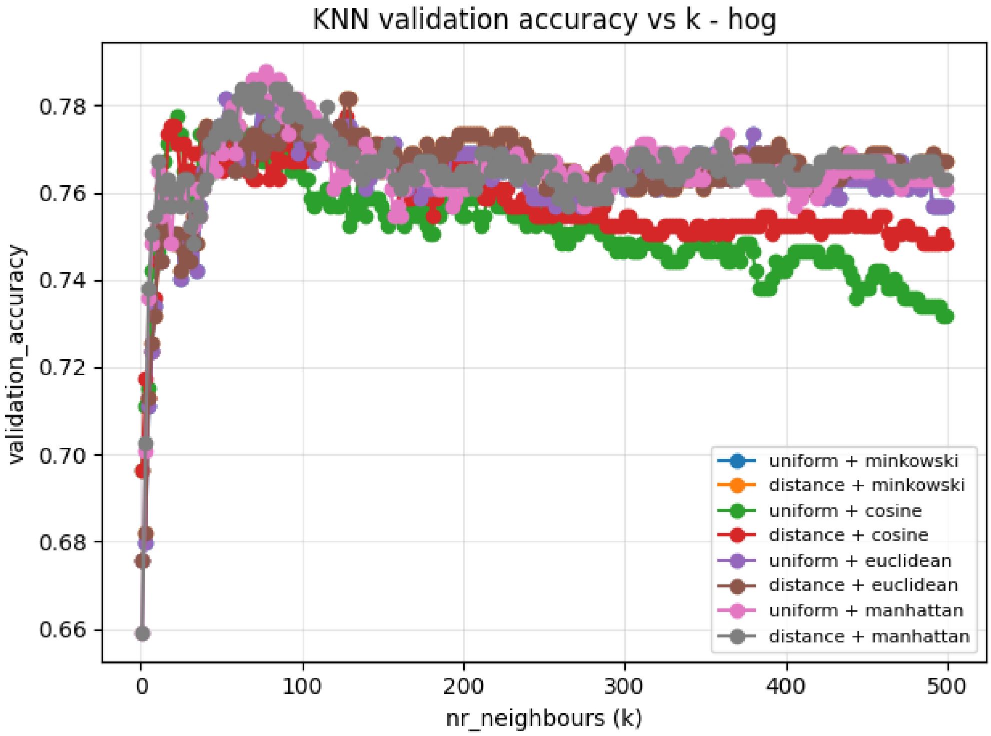

Figure 6: KNN validation accuracy vs. k for HOG features (train/validation only).

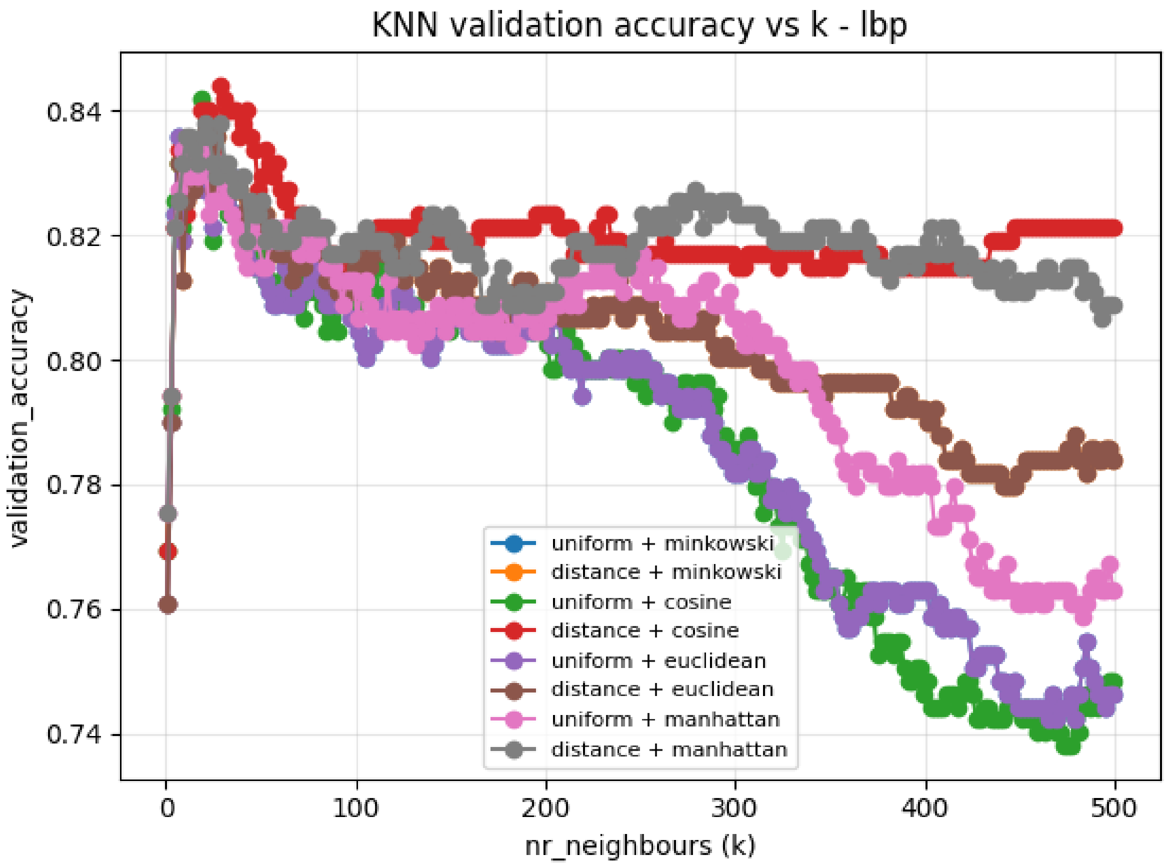

Figure 7: KNN validation accuracy vs. k for LBP features (train/validation only).

In the end, each selected KNN configuration (one for HOG, one for LBP) is retrained on the full training set and evaluated once on the test set.

Hyperparameter	**Value**
Number of neighbors (k)	77
Weighting scheme	uniform
Distance metric	manhattan
Test accuracy	0.7885

Table 2: Best KNN config. - HOG features.

Hyperparameter	**Value**
Number of neighbors (k)	29
Weighting scheme	distance
Distance metric	cosine
Test accuracy	0.8346

Table 3: Best KNN config. - LBP features.

7 Results and interpretation

7.1 What the clusters represent - interpretation

In order to better have an insight of the learned cluster structure, we compute a 2D PCA projection of the feature space after clustering. These visualizations are used just for interpretation.

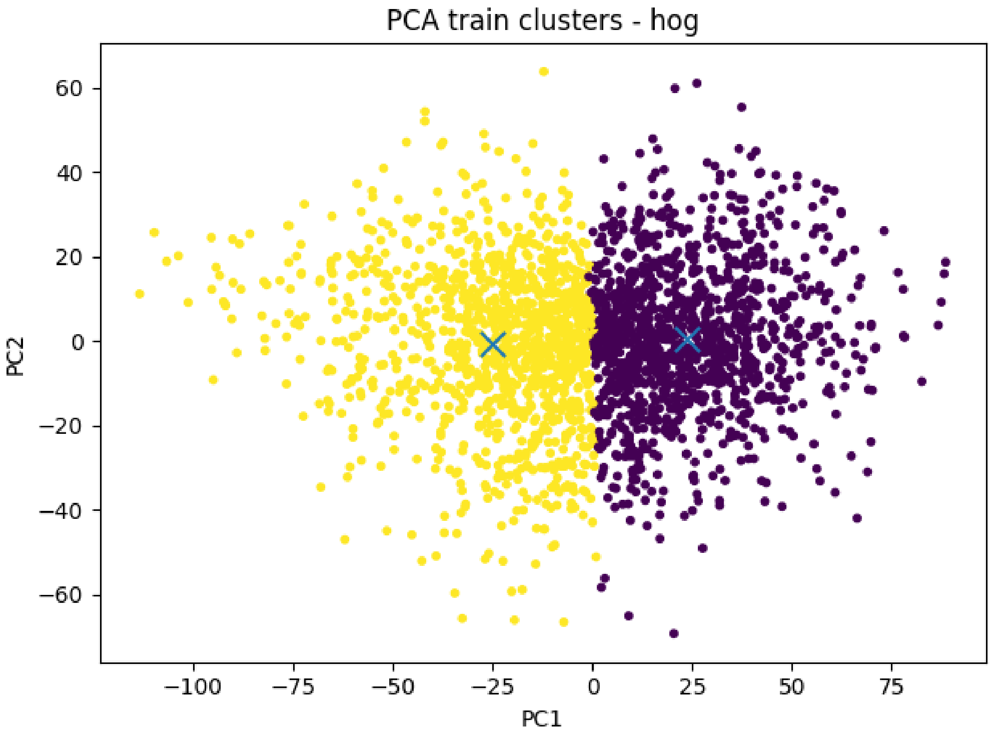

Figure 8: PCA cluster visualization - FCM & HOG. Colors = cluster membership.

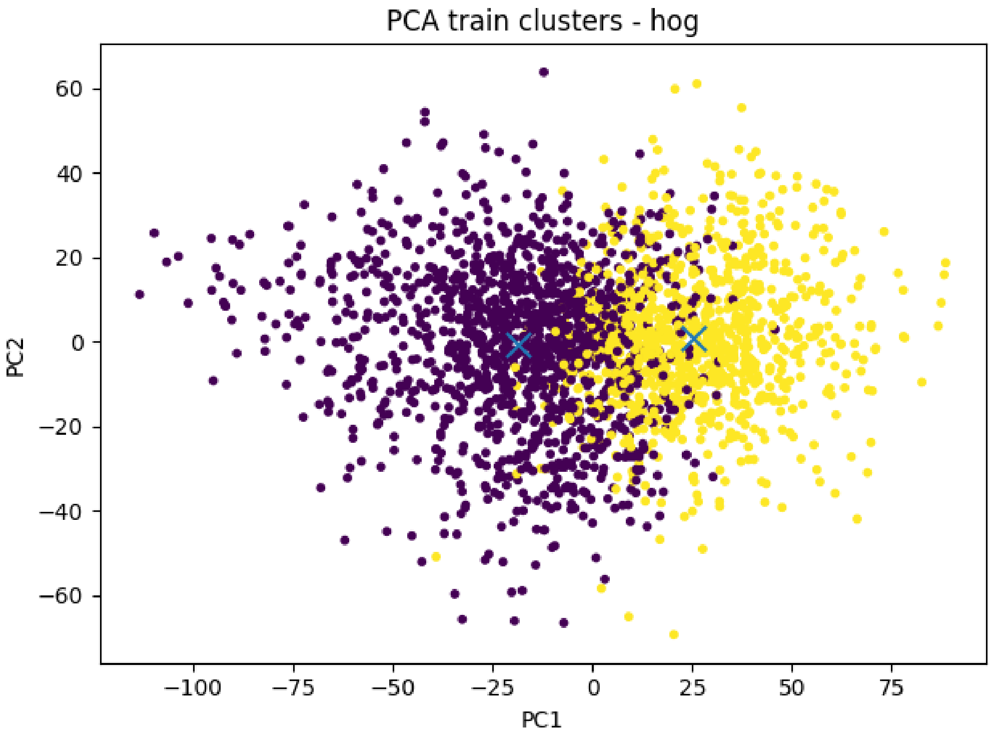

Figure 9: PCA cluster visualization - AHC & HOG. Colors = cluster membership.

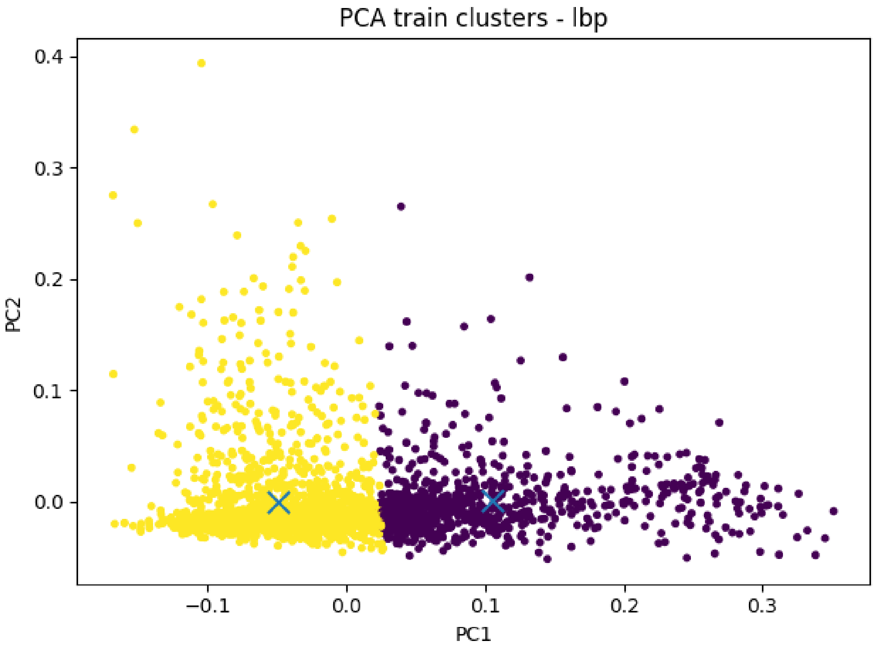

Figure 10: PCA cluster visualization - FCM & LBP. Colors = cluster membership.

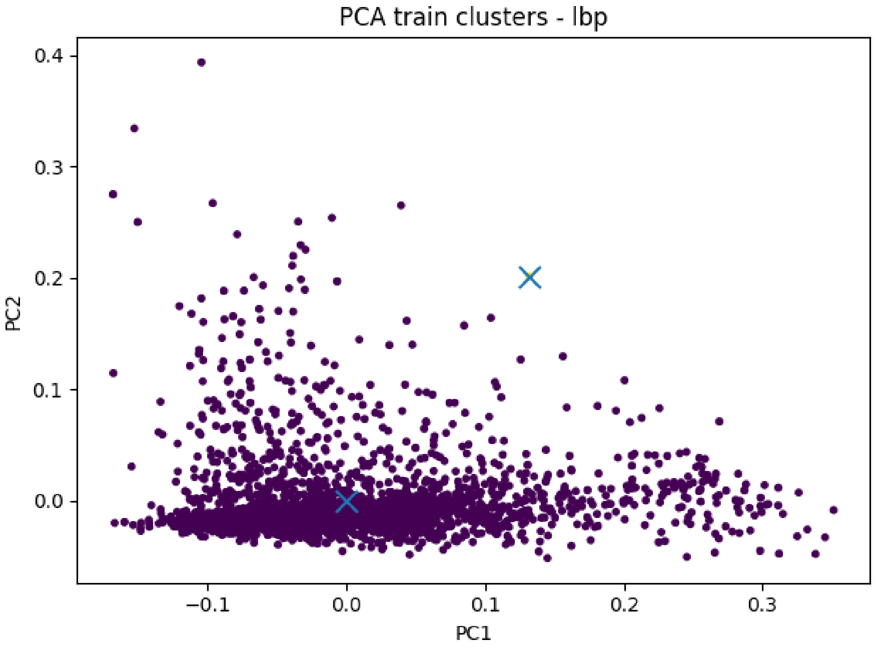

Figure 11: PCA cluster visualization - AHC & LBP. Colors = cluster membership.

We can clearly see a better separation for the HOG-based representations compared to the LBP ones. In both FCM and AHC, HOG features produce relatively compact clusters that are mostly separated along the first principal component.

For the LBP features, on the other hand, the clusters are more elongated and more overlapping in the PCA space. This is expected though, because the LBP encodes local texture statistics rather than full on global structures and the texture patterns of the background, pose changes and illumination of the images introduce more variability that is harder to separate in a low-dimensional linear projection.

Talking about the Agglomerative approach with LBP features, the PCA visualization reveals a degenerate behavior of the clustering, one cluster containing almost all the samples, while the other only has one point. This is a known limitation of single-linkage hierarchical clustering in high-dimensional noisy spaces, the algorithm isolating outliers instead of forming balanced clusters, because of the chaining effects. And while this configuration does obtain a high silhouette score on the training set, the visual inspection shows us that the partition that results does not correspond to an actual meaningful separation of the Horse vs. Zebra domain problem.

On the other hand, the Fuzzy C-Means approach with LBP, produces smoother transitions between the two clusters, although the overall separation remains weaker than the HOG version. That is how we see what the strengths of the HOG features are for this specific problem, but also how important it is to combine quantitative metrics with a qualitative analysis in the evaluation of unsupervised clustering results.

7.2 Quantitative results

This section reports, for each of the 4 pipelines (feature representation × clustering method), the final results on train and test: accuracy, Adjusted Rand Index (ARI), and Normalized Mutual Information (NMI). As stated before, the best configuration is the one selected by the unsupervised criterion on train (FPC for FCM, silhouette for AHC), not by test performance.

Method	Features	Train Acc.	Test Acc.	Train ARI	Test ARI	Train NMI	Test NMI
FCM	HOG	0.6306	0.6654	0.0678	0.1060	0.0522	0.0837
FCM	LBP	0.5339	0.5500	0.0026	0.0060	0.0105	0.0135
AHC	HOG	0.6460	0.6654	0.0846	0.1060	0.0578	0.0837
AHC	LBP	0.5560	0.5500	0.0002	0.0045	0.0010	0.0056

Table 4: Final results for the four experimental pipelines.

Across both of the clustering methods, it seems like the HOG feature representations outperform the LBP ones in terms of accuracy, ARI (Adjusted Rand Index) and NMI (Normalized Mutual Information), meaning that the global shape and edge information is more informative than local texture statistics for separating horses and zebras.

Talking about the HOG features, both Fuzzy C-Means and Agglomerative Hierarchical Clustering achieve similar test accuracies (66.5%), AHC slightly outperforming FCM on the train set. The ARI and NMI values are pretty small, but they are also clearly higher than they are for LBP and indicate a non-random alignment between clusters and true labels. What this means is that HOG captures a meaningful structural separation between horses and zebras, even in an unsupervised setting.

For the LBP feature representations, the performance is close to random chance for both methods, having a test accuracy of $\sim$55% and ARI/NMI being close to 0, meaning that there is a very weak agreement between the clusters and the labels. That seems fair, given that, as it has been stated above, LBP encodes local texture patterns that are heavily affected by background variability, pose and illumination, which make global separation harder to achieve without supervision.

Comparing the two clustering methods, we safely say that the Agglomerative approach slightly outperforms the Fuzzy algorithm on HOG features, but that is not the case for the LBP ones. The really low values for ARI and NMI for AHC with LBP shows the degenerate clustering behavior we also observed in the PCA plots. That way, we can observe that a high unsupervised score (like the silhouette) does not necessarily guarantee us a meaningful partition for our problem.

8 Conclusion

The scope of this project was to explore unsupervised separation on the Horse2zebra dataset, using, as algorithms, Fuzzy C-Means and Agglomerative Hierarchical Clustering. The results of the experiments have clearly shown that the **Histogram of Oriented Gradients** outperforms the Local Binary Patterns by a significant margin, having a peak accuracy on the test subset of **66.54%** with both clustering methods.

While it is true that the unsupervised models successfully exceed the random baseline (50%), they remain inferior to the supervised KNN model (∼83%), the analysis confirming that for this specific task, the global structural information like the distinct edges of zebra stripes are a far better feature space than local texture statistics. Furthermore, the degenerate behavior observed in LBP clustering highlights that high internal validation scores (like the silhouette) do not always guarantee a meaningful semantic separation of the data.

References

[1] Berkeley AI Research (BAIR), *Horse2zebra dataset (cyclegan)*, https://www.kaggle.com/datasets/balraj98/horse2zebra-dataset, Originally introduced with CycleGAN, 2017.