Comparative Study of Classification Algorithms

AID Dataset Classification Using Bag of Features
Machine Learning

Javier González Villasmil
Claudia Élez Mencía

November 2025
Sapienza University of Rome

Contents

1 Introduction and motivation

Image classification is a fundamental task in computer vision and plays a fundamental role in many perception tasks related to Robotics. The Aerial Image Dataset (AID)[1] is composed of high-resolution scenes from Google Earth imagery captured via aerial platforms or satellites.

As students in the robotics software field, we consider this dataset in particular appropriated because aerial image is commonly used in autonomous systems such as UAVs in multiple tasks such as navigation, mapping, or monitoring, among others.

Since traditional classifiers require vectorial numerical features, we will try the Bag of Features (BoF) approach, which consist in separate the image in different features (by descriptors) to obtain an histogram.

We will also evaluate different descriptors, in order to find the most optimal for the task. Once we have obtained the histograms or vectors that contains the images features, we train and compare different multi-class classifiers, such as Naïve Bayes, Softmax Regression, Random Forest among others. We also included a Convolutional Neural Network (CNN) to contrast the performance of Classical Machine Learning methods against a Deep Learning Approach.

2 Dataset Description

The AID dataset[1] contains 10,000 aerial images with typical resolution of 600×600 pixels, grouped into the following 30 classes:

Airport, bare land, baseball field, beach, bridge, center, church, commercial, dense residential, desert, farmland, forest, industrial, meadow, medium residential, mountain, park, parking, playground, pond, port, railway station, resort, river, school, sparse residential, square, stadium, storage tanks and viaduct.

The dataset is approximately balanced, with around 200–400 images per category. We perform a 90/10 split into training and test sets. For the validation, we used the k-fold cross validation to tune the hyperparameters, which consists in divide the dataset into k equal partitions, for each iteration, one fold is used for validation and the remaining ones for training. We decided to use a standard k=5, which means that it separates the training data into 80% training set and 20% validation set

3 Methodology and Models

Since there are no labels missing, and images do not require denoising, the only necessary pre-processing concerns feature extraction, for that purpose, we will use the Bag of Features approach.

Bag of Features (BoF). The Bag of Features (BoF) model [2], also known as the Bag of Visual Words (BoVW), is a classical image representation technique that converts an image into a fixed-length numerical vector.

This process is implemented in 4 steps

1. Determination of Images Features of a given label, using descriptors.
2. Construction of visual vocabulary, by clustering.
3. Image Encoding.
4. Classification.

3.1 Local Descriptors Evaluated

A local descriptor extracts information from small regions of an image and represents it as a numerical vector that characterizes the visual features of detected keypoint and encode the appearance of local patches and provide robustness to changes in scale, rotation and illumination, making them suitable for classical image representation pipelines.

In this study, we evaluate these three widely used local feature descriptors: SIFT, SURF and ORB.

ORB. Oriented FAST and Rotated BRIEF ORB is a binary descriptor designed for speed. It extracts thousands of features quickly, but its descriptive power is limited. In our experiments, ORB-based BoF histograms showed high intra-class variability, resulting in poor classification performance.

SURF. Speeded Up Robust Features SURF is more descriptive and robust, but its computational cost is significantly higher. SURF is a proprietary local feature and descriptor detector so it can't be used commercially

SIFT. Scale Invariant Feature Transform SIFT provides highly robust descriptors with strong invariance to scale, rotation and illumination changes. Although it is usually slower than SURF, extraction was feasible on our hardware and produced substantially better results. Therefore, SIFT is selected for the final BoF representation.

AKAZE. AKAZE offers nonlinear scale-space detection and is more robust than ORB. However, in our experiments its performance was comparable or slightly inferior to SIFT, however it does computations faster.

3.2 Clustering

K-MiniBatchMeans K-MiniBatchMeans is a clustering method based in K-Means.

Both clustering methods group similar descriptors and find k centroids that represent common visual patterns (image features), the difference is that K-Means uses all the descriptors at each iteration, while MiniBatch K-Means use a batch composed with a reduced number of descriptor.

In this case, after trying several values for k, we set k=256, because greater values supposed a larger amount of time.

3.3 Image Encoding

Bag Of Features (BoF). The BoF approach provides the histogram of the assigment of all image descriptors to visual words.

Vector of Locally Aggregate Descriptors (VLAD). VLAD representation [3] aggregates descriptors based on a locality criterion in feature space. Each local descriptor x is associated to its nearest visual word $\mathbf{c}_i$=NN(x). The idea of the VLAD descriptor is to accumulate, for each visual word $\mathbf{c}_i$, the differences x$\mathbf{c}_i$ of the vectors x assigned to $\mathbf{c}_i$. This characterizes the distribution of the vectors with the features respect to the center.

3.4 Machine Learning Classification Models

We evaluate the following multi-class classification models:

- **Gaussian Naïve Bayes.** This classifier is based on the Bayes Theorem with the assumptions of conditional independence and Gaussian distributions for each feature, for a a feature vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, the posterior probability of class C_k is:

  $$P(C_k \mid \mathbf{x}) \propto P(C_k) \prod_{i=1}^{n} P(x_i \mid C_k).$$

  with the following likelihood:

  $$P(x_i \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_{ki}^2}} \exp\left(-\frac{(x_i - \mu_{ki})^2}{2\sigma_{ki}^2}\right).$$

  Then, the predicted class will be:

  $$\hat{C} = \arg\max_{C_k} \; P(C_k) \prod_{i=1}^{n} P(x_i \mid C_k).$$

- **Softmax Regression.** It is a generalized version of the Logistic Regression for multi-class classification. We compute a linear score for each class:

  $$\hat{p}_k(x) = \frac{e^{s_k(x)}}{\sum_\ell e^{s_\ell(x)}}$$

  With the following log-likelihood function:

  $$\hat{\ell}(\Theta) = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} y_i^{(k)} \log\left(\hat{p}_k(x_i)\right)$$

  The predicted label corresponds to the class with the highest estimated probability.

- **Decision Tree.** This model consists in splitting the data into smaller, more homogeneous subsets, by conditions. At each node, the algorithm chooses the feature and threshold to best separate the classes. The process continues until a stopping condition is met.

  This gives as a result a node tree, where the leaves are the final predicted values.

- **Random Forest.** It is based on the Decision Tree Model, instead of training only one Tree Predictor, it trains K shorter trees.

- **Support Vector Classifier (SVC, Kernel-Based)** This Classification Model is a version of the Support Vector Machine, whose purpose is to find an hyperplane that separates the classes with the maximum margin possible. We used a Linear Kernel

3.5 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a special type of feed-forward neural network that use convolutions, a mathematical operation that applies small filters across the image and computes weighted sums over local regions. These filters learn to detect meaningful visual patterns, such as edges, corners, textures, and parts of objects. Because convolutions reuse the same weights across the image, CNNs are naturally more robust to translations, small rotations, and local distortions—problems where classic feature extractors and Bag-of-Visual-Words (BoVW) pipelines typically struggle.

This makes CNN-based image classification simpler and more effective. They eliminate all BoVW steps (keypoint detection, descriptor extraction, clustering, encoding, and classification), replacing them with a single end-to-end model that learns the features directly from data. Although CNNs are theoretically more complex internally (since we don't fully understand the neural network), their implementation is far simpler and they generally offer higher accuracy and greater robustness to variations in the input.

To further improve performance, we incorporated residual connections, which allow the gradient to flow more easily during training and prevent degradation in deeper networks. Residual blocks enable deeper feature extraction without significantly increasing computational cost—only a slight increase in model size and training time.

Our architecture consists of an initial convolution layer followed by three residual blocks. Each block contains convolutional layers, batch normalization (to stabilize learning), and max-pooling (to reduce spatial dimensions and lower the risk of overfitting). Using this lightweight configuration of around 1 million parameters, we achieved approximately 92% test accuracy, training for around 2 hours on an 8 GB GPU. Note that the final model simply occupies a mere 16 MB of size

4 Results and Analysis

4.1 Quantitative Metrics

Table 1: Model performance using SIFT-based BoF features.

Model	Train Accuracy	Test Accuracy	Precission	Recall	F1-score	Overfit
NB	0.5804	0.5080	0.5108	0.5086	0.4982	0.0724
Softmax	0.7344	0.6270	0.6119	0.6177	0.6060	0.1074
DT	0.6489	0.2755	0.2960	0.2741	0.2783	0.3734
RF	0.9996	0.5380	0.5347	0.5261	0.5019	0.4616
SVC (L)	0.9959	0.6835	0.6716	0.6781	0.6715	0.3124
SVC (RBF)	0.9996	0.6940	0.6883	0.6896	0.6843	0.3056

About the Convolutional Neural Network, we obtained the following metrics:

- Accuracy: 92.80%
- Macro Average: 0.93
- Weighted Average: 0.93

4.2 ROC Curves

A ROC curve for a multi-class classifier is obtained by calculating the True Possitive Rate (TPR) and the False Posistive Rate (FPR), it is mapped between 0 and 1. As it gets closer to one, more precision the model has.

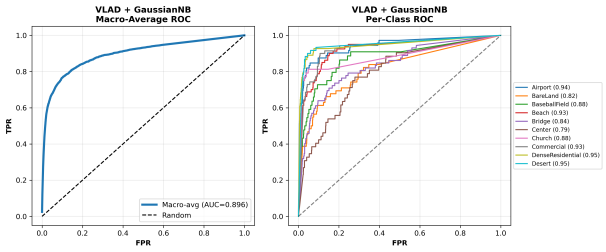

Figure 1: Naïve Bayes ROC Curve

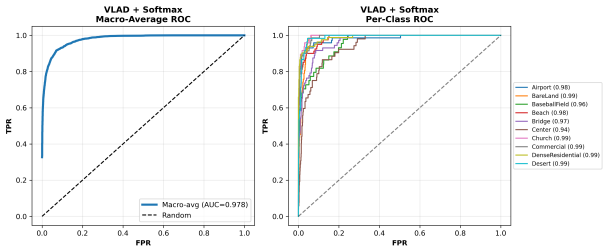

Figure 2: Softmax ROC Curve

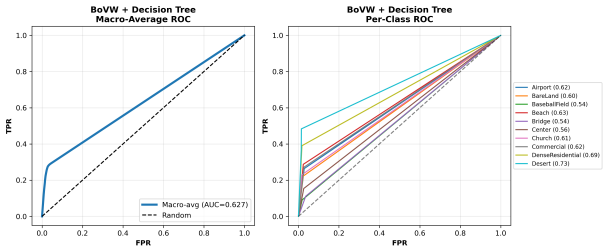

Figure 3: Decision Tree ROC Curve

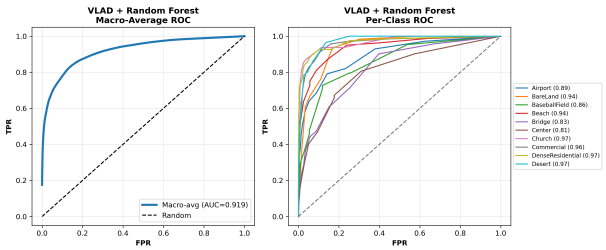

Figure 4: Random Forest ROC Curve

4.3 Confusion Matrix

A Confusion Matrix for multi-class classifiers, consist of a squared matrix where the columns represent the predicted classes and the rows the actual classes. Each cell shows the number of

instances where an actual class was predicted as another class. Correct predictions are on the diagonal, while off-diagonal cells show misclassifications.

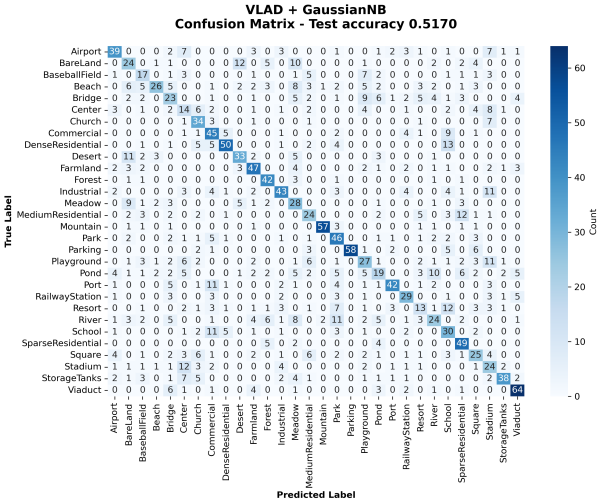

Figure 5: Naïve Bayes Confusion Matrix

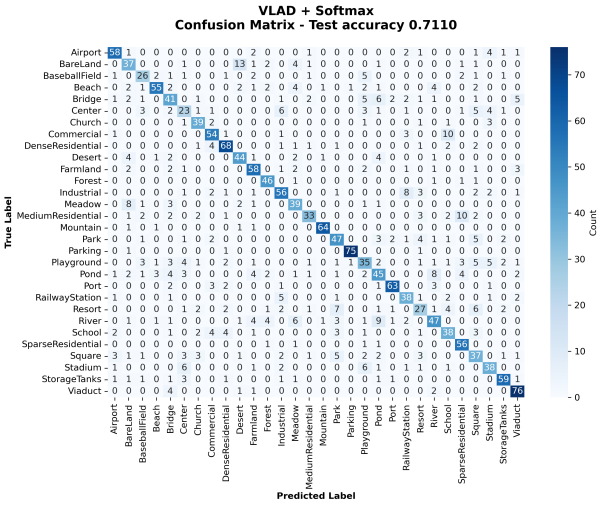

Figure 6: Softmax Confusion Matrix

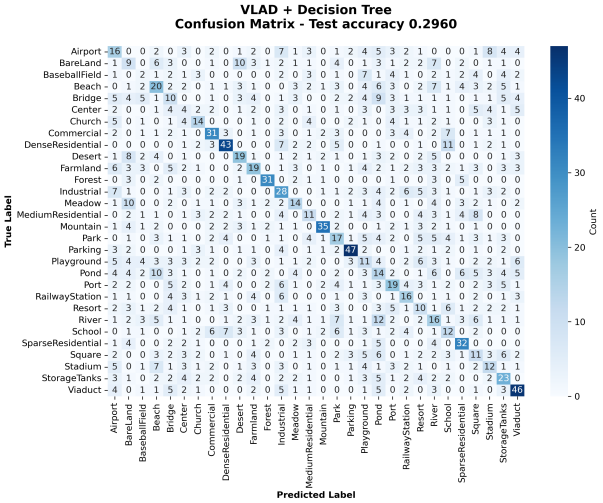

Figure 7: Decision Tree Confusion Matrix

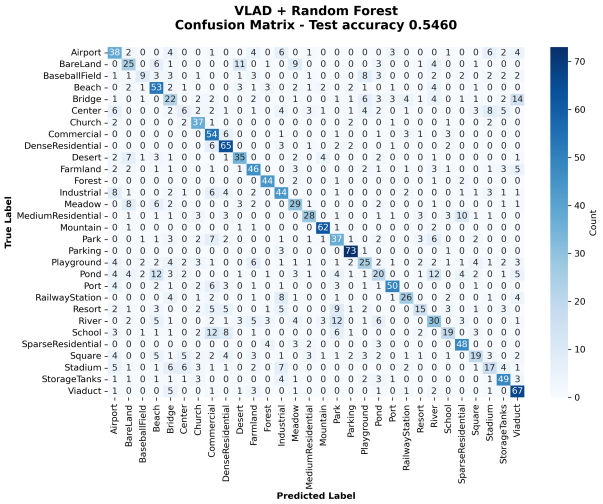

Figure 8: Random Forest Confusion Matrix

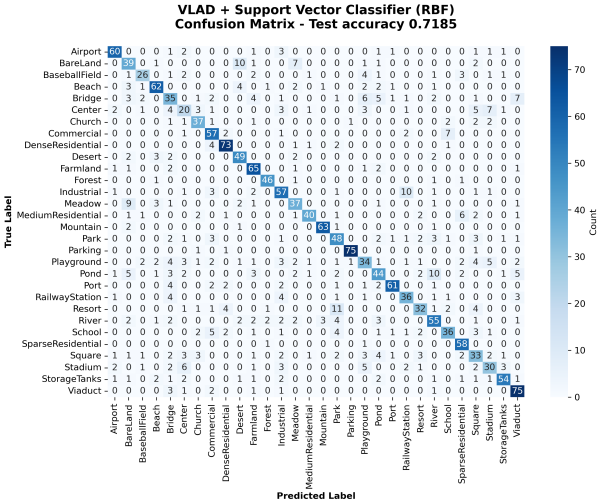

Figure 9: Support Vector Machine Confusion Matrix

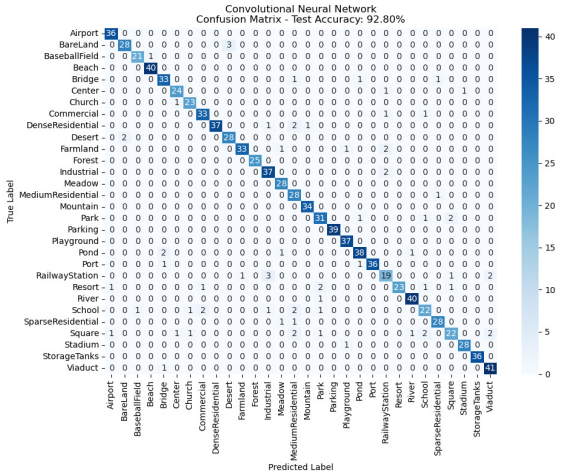

Figure 10: CNN Confusion Matrix

4.4 Discussion

4.4.1 Descriptors evaluation

After some trials, we can ensure that ORB, despite of being the fastest one, produced noisier histograms, which harmed the classifiers performance.

SURF could not be used due to license limitations that cannot be used with the standard OpenCV library.

AKAZE provided considerably better performance than the ORB, while being faster and lighter than SIFT. It is a good middle-ground detector to be considered in some scenarios

Finally, we choose the SIFT descriptors, as we seen have proved that they provide much better separability in the BoF feature space compared to the ORB and the AKAZE.

4.4.2 Classificators evaluation

SVM with RBF kernel achieves the best performance due to its ability to learn nonlinear boundaries. Decision Trees overfit, Random Forest improves robustness, and Logistic/Softmax Regression struggle in complex class boundaries.

First, we can observe in the metrics (Table 1) the significant presence of overfitting in the Decision Tree and Random Forest models. They have great results at training (0.68, 0.99 respectively), but poor test accuracy (0.27 and 0.53). This means that they "memorized" the training set.

The Naïve Bayes model achieves a precision of the 0.51, this is due to the assumptions of independence of the model. In addition, the F1-Score reached is 0.4982, this shows that this classifier have a significant number of errors in its predictions.

We can also observe that the Softmax model has a reasonable performance, with an accuracy of 0.7344 for the training and 0.627 for the test. It is also the classifier with the less overfitting indicator.

About the Machine learning methods, the best performances have been made by the Support Vector Classifier, both the Linear and the RBF versions reach a 0.68 - 0.69 test accuracy. The F1-scores in both cases represent that this models are the more stable ones. The RBF SVC has a slightly improve from the Linear one. On the other hand, they both present a considerable percentage of overfitting.

Finally, observing the accuracy obtained by the CNN and its Confusion Matrix, we can ensure that the best performance has been reached by the Convolutional Neural Network, with a total accuracy of the 92%, when the best results in the ML Classifiers were around the 72% in their test accuracy.

5 Conclusion

Using a Bag of Features representation with SIFT descriptors, allows some of the Classical Machine Learning Processes, Softmax and Support Vector Classification, to obtain great performances on the AID Dataset. Nevertheless, the Convolutional Neural Networks reaches more optimal results in faster times.

References

[1] AID: A Scene Classification Dataset. Available at: https://www.kaggle.com/datasets/jiayuanchengala/aid-scene-classification-datasets/data

[2] IBM, "Bag of Words," IBM Think Blog. Available at: https://www.ibm.com/think/topics/bag-of-words

[3] H. Jégou, M. Douze, C. Schmid and P. Pérez, "Aggregating local descriptors into a compact image representation," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010, pp. 3304-3311, doi: 10.1109/CVPR.2010.5540039. Available at: https://ieeexplore.ieee.org/document/5540039

[4] Scikit-Learn Documentation. https://scikit-learn.org