

Classification de CIFAR

Vincent TARDIEUX — Joris LHOMME

Contents

1	Introduction et état de l'art	1
1.1	Présentation du dataset CIFAR	1
1.2	Etat de l'art	1
1.3	Lien du dépôt Github	1
2	Benchmarking	1
2.1	Pré-traitement	1
2.2	Extraction d'attributs	1
2.2.1	Vecteurs aplatis	1
2.2.2	Local Binary Pattern (LBP)	2
2.2.3	Histogram of Gradient (HoG)	2
2.3	Scale-Invariant Feature Transform (SIFT)	3
2.3.1	benchmarking des méthodes d'extraction	4
2.4	Réduction de dimension et sélection d'attributs	4
2.4.1	Réduction de dimension	4
2.4.2	Sélection d'attributs	5
2.5	Comparaison et sélection du meilleur modèle de classification	5
2.6	Optimisation des hyperparamètres du meilleur modèle	5
2.6.1	Validation croisée	5
2.6.2	GridSearch	5
3	Conclusion et Perspectives	6
3.1	Conclusion	6
3.2	Perspectives	6
4	Annexe	7

1 Introduction et état de l’art

1.1 Présentation du dataset CIFAR

Dans la suite de ce rapport nous utiliserons le dataset [CIFAR](#).

L’objectif de ce projet est de classifier des images appartenant à l’une des dix classes suivantes :

- Avion
- Automobile
- Oiseaux
- Chat
- Biche
- Chien
- Grenouille
- Cheval
- Bateau
- Camion

Il est intéressant de noter que ces dix classes sont équilibrés [\[5\]](#)

1.2 Etat de l’art

En se basant sur [PapersWithCode](#), la meilleure précision observée sur CIFAR-10 est à 99.612% avec le modèle [Efficient Adaptive Ensembling for Image Classification](#), lorsque l’on abandonne l’apprentissage profond on tombe vers les 60% de précision.

1.3 Lien du dépôt Github

Le code source de notre bibliothèque et son installation est accessible [ici](#).

2 Benchmarking

2.1 Pré-traitement

Parmi les différents pré-traitements que l’on a essayé, quasiment aucun n’ont permis l’amélioration de résultats. Par exemple l’application de Grayscale (conversion de l’image en nuances de gris) fait perdre trop d’informations sur nos images. Nous pensons qu’avec des images plus grandes (par exemple 256x256 au lieu de 32x32) l’application de telles opérations auraient plus de sens.

Finalement le pré-traitement qui a été conservés est uniquement une normalisation de nos images.

2.2 Extraction d’attributs

Pour l’extraction d’attributs, quatre méthodes ont été essayés :

- Vecteurs aplatis
- Local Binary Pattern (LBP)
- Histogram of Gradient (HoG)
- Scale-Invariant Feature Transform (SIFT)

2.2.1 Vecteurs aplatis

L’idée de la méthode des vecteurs aplatis est de transformer l’image en un unique vecteur de pixel que l’on donne en tant que vecteur d’attributs à notre algorithme de classification. Cette méthode nous servira surtout de base de comparaison avec les autres méthodes d’extraction d’attributs.

2.2.2 Local Binary Pattern (LBP)

L'algorithme du Local Binary Pattern (LBP) extrait des caractéristiques de texture en comparant chaque pixel d'une image à ses voisins. Pour chaque pixel, les comparaisons sont codées en motifs binaires, convertis en valeurs décimales, puis utilisés pour construire un histogramme des motifs locaux. Cet histogramme sert de vecteur d'attributs.

Le LBP est utilisé pour des tâches comme la reconnaissance faciale et l'analyse de textures.

Le LBP a été très vite abandonnée puisque encore une fois, nos images ont une résolution trop faible pour différencier des textures. [1]

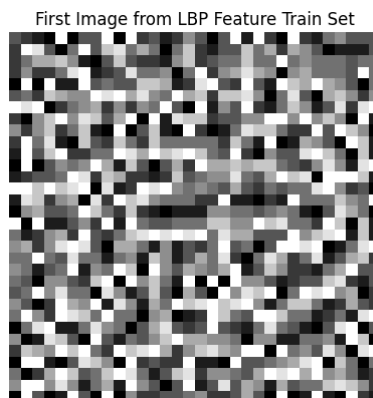


Figure 1: Exemple d'image de sortie du LBP

2.2.3 Histogram of Gradient (HoG)

La méthode Histogram of Gradients (HoG) extrait des caractéristiques en analysant les gradients de luminosité locaux. L'image est divisée en cellules, et pour chaque cellule, un histogramme des orientations des gradients est calculé. Ces histogrammes sont ensuite normalisés et combinés pour former un vecteur d'attributs. HoG est utilisé pour des tâches telles que la détection d'objets et la reconnaissance de formes.

[2]

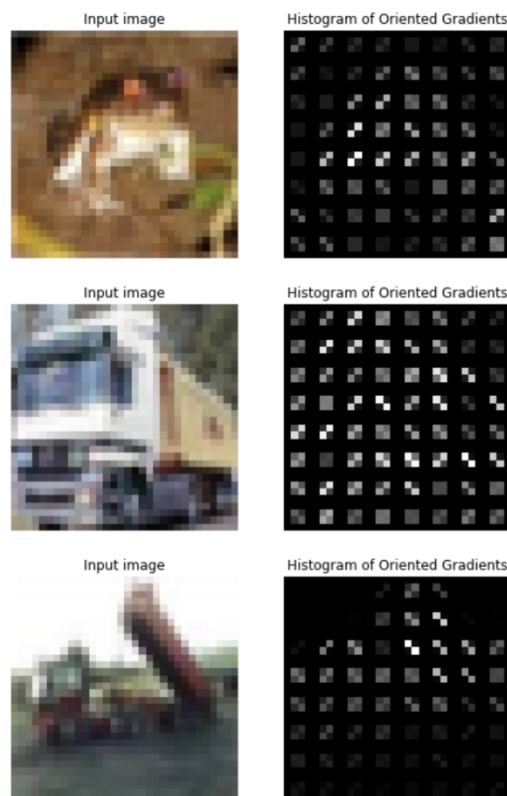


Figure 2: Exemples d'images sur lesquelles on applique la méthode HoG

2.3 Scale-Invariant Feature Transform (SIFT)

La méthode Scale-Invariant Feature Transform (SIFT) détecte et décrit des points d'intérêt invariants à l'échelle et à la rotation. Il identifie des points clés dans l'image, calcule leurs descripteurs basés sur les gradients locaux, et crée un vecteur d'attributs pour chaque point clé. SIFT est utilisé pour des tâches comme la reconnaissance d'objets et le suivi de mouvements.[3]

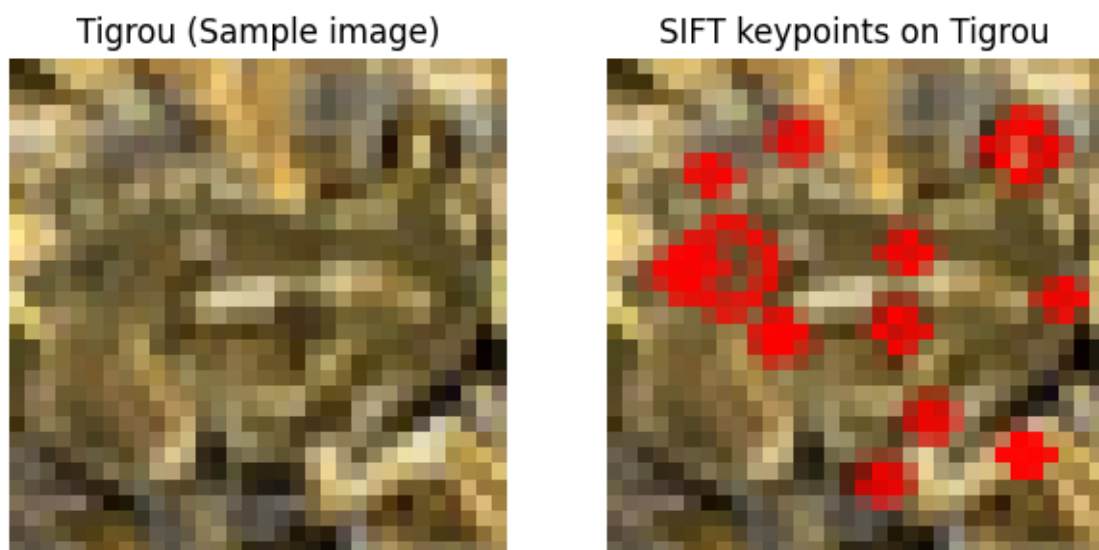


Figure 3: Exemple d'image sur laquelle on applique la méthode SIFT

2.3.1 benchmarking des méthodes d'extraction

Lors de la comparaison des trois extracteurs d'attributs restants la méthode HoG se démarque lorsque l'on teste avec différents modèles. On obtient par exemple une précision de 20% avec SIFT contre 50% avec HoG et 45% avec les vecteurs aplatis en utilisant le modèle de régression logistique. Nous garderons donc par la suite uniquement la méthode HoG.

2.4 Réduction de dimension et sélection d'attributs

En sortie de nos méthodes d'extraction d'attributs, nos vecteurs peuvent être plutôt volumineux, on souhaite donc les réduire en utilisant deux types de méthodes distinctes : la réduction de dimension et la sélection d'attributs.

2.4.1 Réduction de dimension

L'analyse de composant principal (PCA) est une technique de réduction de dimensionnalité qui transforme les données en un nouvel espace de caractéristiques. Elle identifie les directions (composantes principales) où les données varient le plus et projette les données dans cet espace en conservant autant de variance que possible. La PCA permet de simplifier les données tout en minimisant la perte d'information, facilitant ainsi l'analyse et la visualisation.

Après quelques tests nous avons choisi arbitrairement de faire passer la dimension des descripteurs de SIFT de 128 dimensions à 64. Pour les autres méthodes les résultats ont drastiquement baissé, nous avons donc décidé de ne pas l'utiliser.^[4]

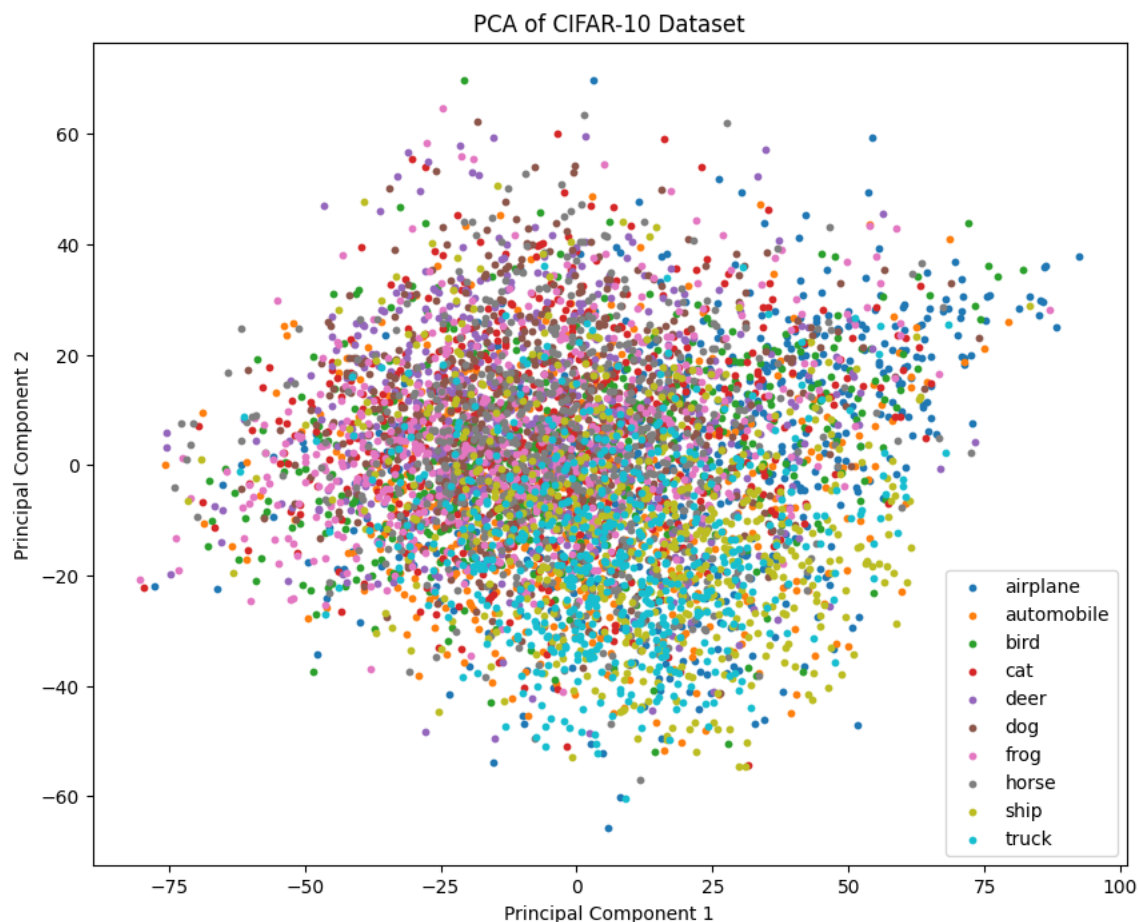


Figure 4: 2 meilleurs composants d'une PCA avec une dimension de sortie à 64 sur les vecteurs aplatis

2.4.2 Sélection d'attributs

La sélection d'attributs consiste à identifier les pixels dont la variance des valeurs est significative par rapport à un seuil prédéfini ("Threshold" dans notre code). Cette méthode filtre les pixels peu informatifs, améliorant ainsi l'efficacité des algorithmes en se concentrant sur ceux qui contribuent le plus à la différenciation des caractéristiques visuelles.

Encore une fois, l'application de cette méthode a fait chuté les résultats, nous prenons donc la décision de ne pas l'utiliser par la suite.

2.5 Comparaison et sélection du meilleur modèle de classification

Nous avons choisi 6 classifieurs différents :

- SVC
- Logistic Regression
- KNN
- Random Forest
- Decision Tree
- Gradient Boosting

Après avoir effectué des tests avec chaque méthode d'extraction d'attribut on observe une meilleure précision avec le classifieur SVC (Voir Table [1])

2.6 Optimisation des hyperparamètres du meilleur modèle

Maintenant que l'on a pu désigner notre meilleur modèle (à savoir, le SVC) nous allons chercher à optimiser ses hyperparamètres afin d'améliorer son score de précision.

2.6.1 Validation croisée

La validation croisée consiste à diviser les données en plusieurs sous-ensembles. Le modèle est entraîné sur certains de ces sous-ensembles et testé sur les autres. Ce processus est répété plusieurs fois en changeant les sous-ensembles de test et d'entraînement à chaque itération. Les performances obtenues sont ensuite moyennées pour obtenir une estimation plus fiable et robuste de l'efficacité du modèle.

A l'aide de la validation croisée on a pu confirmer que notre modèle n'a pas overfitté et possède une faible variance avec une précision moyenne de 58% sur 6 folds (6 itérations).

2.6.2 GridSearch

Le GridSearch consiste à explorer systématiquement un espace de combinaisons d'hyperparamètres spécifié par l'utilisateur, en entraînant et en évaluant le modèle pour chaque combinaison. Les performances sont mesurées à l'aide de techniques comme la validation croisée, et la combinaison d'hyperparamètres qui donne les meilleurs résultats est sélectionnée pour le modèle final.

Pour le SVC les différents hyperparamètres sur lesquels nous pouvons jouer sont :

- C : Paramètre de régularisation qui contrôle la pénalité pour les erreurs de classification, avec une valeur plus élevée indiquant une régularisation plus forte.
- gamma : Coefficient du noyau pour les noyaux 'rbf', 'poly', et 'sigmoid', déterminant l'influence de chaque exemple de formation; une valeur plus faible signifie une influence plus grande.
- kernel : Type de noyau à utiliser dans le SVC, permet d'influencer la flexibilité du modèle dans la classification des données.
 - **'rbf' (Radial Basis Function)** : Utilise un noyau gaussien pour transformer les données en un espace à dimension infinie, adapté aux données non linéairement séparables.
 - **'poly' (Polynomial)** : Utilise un noyau polynomial pour transformer les données en un espace de grande dimension, efficace pour capturer les relations non linéaires entre les caractéristiques.

- **'sigmoid'** : Utilise un noyau sigmoïdal pour transformer les données, similaire à une fonction d'activation sigmoïde en réseaux de neurones, adapté à des distributions de données spécifiques et moins courant que les noyaux 'rbf' et 'poly'.

Notre grille de paramètre choisit est la suivante :

- 'C': [0.1, 1, 10, 100]
- 'gamma': [1, 0.1, 0.01, 0.001, 'scale']
- 'kernel': ['rbf', 'poly', 'sigmoid']

On obtient les hyperparamètres optimaux suivant **C = 10**, **Gamma = 'scale'**, **kernel = 'rbf'** avec une précision moyenne de 55.16%, ce qui est plus bas que les hyperparamètres par défaut, nous pensons que cela vient du fait que lors de la GridSearch, il y a une validation croisée qui s'effectue implicitement, moyennant les précisions et donc la réduisant.

3 Conclusion et Perspectives

3.1 Conclusion

Dans ce projet, nous avons exploré différentes méthodes pour classifier des images du dataset CIFAR-10, une base de données bien connue dans le domaine de la vision par ordinateur. Notre objectif était de comparer différentes techniques d'extraction d'attributs et divers modèles de classification pour identifier la combinaison la plus efficace.

Nous avons commencé par le pré-traitement des images, où nous avons trouvé que la normalisation était la seule méthode qui améliorait les résultats. Ensuite, nous avons testé plusieurs méthodes d'extraction d'attributs, notamment les vecteurs aplatis, Local Binary Pattern (LBP), Histogram of Gradients (HoG), et Scale-Invariant Feature Transform (SIFT). Parmi ces méthodes, HoG s'est démarqué en fournissant les meilleurs résultats.

Nous avons également exploré des techniques de réduction de dimension et de sélection d'attributs, mais celles-ci n'ont pas amélioré les performances de manière significative. Par conséquent, nous avons décidé de ne pas les utiliser dans notre pipeline final.

Pour la classification, nous avons comparé plusieurs modèles, dont SVC, Logistic Regression, KNN, Random Forest, Decision Tree, et Gradient Boosting. Le modèle SVC a montré les meilleures performances avec une précision moyenne de 58%. Ensuite nous avons effectué l'optimisation des hyperparamètres via GridSearch, nous avons essayé d'affiner les paramètres du SVC, obtenant une précision finale de 55.16% avec les paramètres optimaux.

3.2 Perspectives

Bien que nous ayons atteint des résultats satisfaisants, plusieurs avenues peuvent être explorées pour améliorer davantage la performance du modèle :

- **Augmentation des données**: Utiliser des techniques d'augmentation des données pour générer des images supplémentaires à partir des images existantes, ce qui pourrait aider à mieux généraliser le modèle. (E.G: Appliquer des rotations ou d'autres transformations sur les images du jeu de donnée)
- **Utilisation de réseaux de neurones profonds**: Étant donné les limitations des méthodes classiques de machine learning sur des données complexes comme CIFAR-10, l'utilisation de réseaux de neurones convolutifs (CNN) pourrait considérablement améliorer les performances de classification.
- **Ensembles de modèles**: Combiner plusieurs modèles de classification en un ensemble pourrait fournir des prédictions plus robustes et précises.

En conclusion, bien que notre approche actuelle offre une base solide pour la classification d'images CIFAR-10, il existe de nombreuses opportunités pour améliorer et affiner notre modèle en explorant des techniques plus avancées et en intégrant des approches de l'apprentissage profond.

4 Annexe

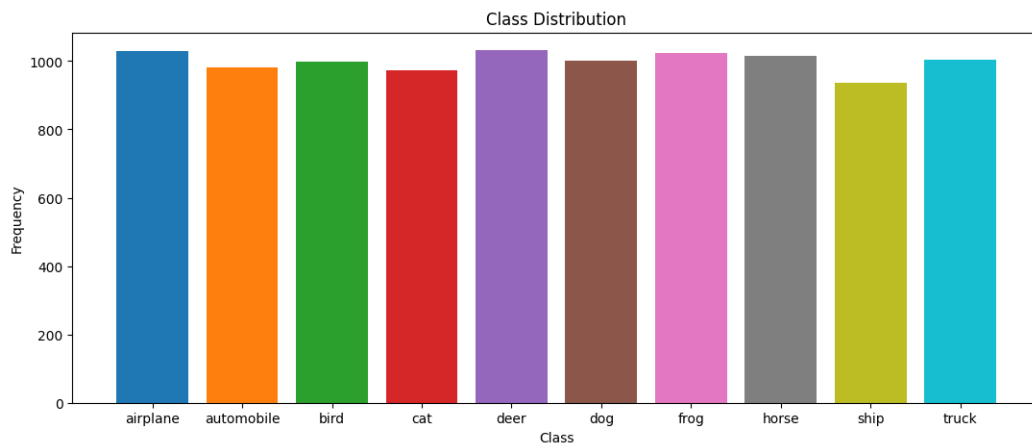
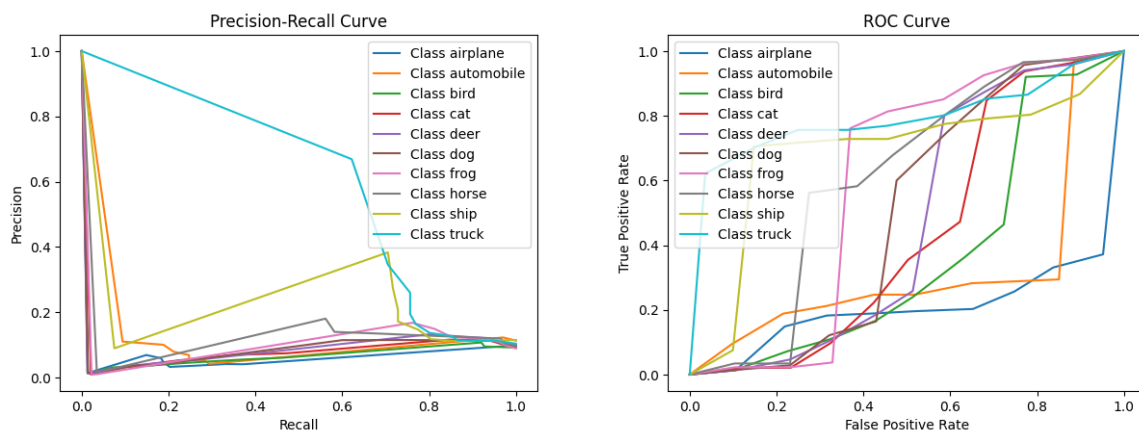


Figure 5: Distribution des classes à travers le jeu de données

	SVC	Log. Reg.	Rand. For.	Dec. Tree	KNN	Grad. Boost.
HoG	0.58	0.52	0.48	0.25	0.42	0.52
Flat	0.48	0.29	0.42	0.24	0.30	0.45
SIFT	0.32	0.26	0.19	0.20	0.25	0.30
LBP	0.15	0.10	0.09	0.08	0.13	0.15

Table 1: Comparaison des différents classifieurs avec des hyperparamètres par défaut et les vecteurs HoG



(a) Courbe de précision / rappel de SVC avec les hyperparamètres optimaux et vecteurs HoG (b) Courbe ROC de SVC avec les hyperparamètres optimaux et vecteurs HoG

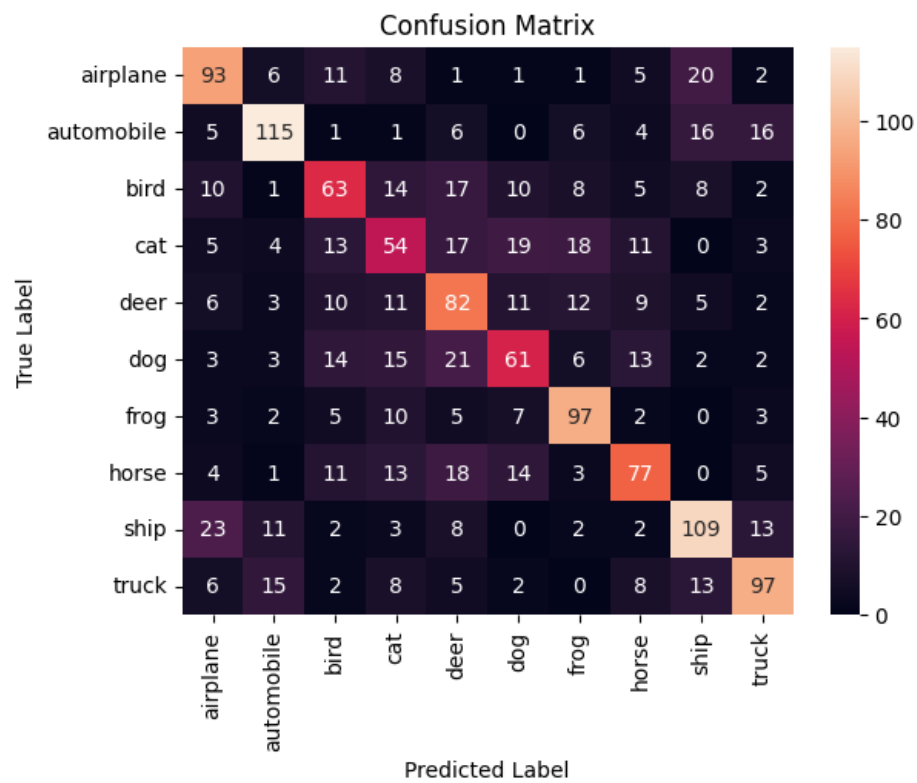


Figure 7: Matrice de confusion de SVC avec les hyperparamètres optimaux et vecteurs HoG