

# Classification d'images avec SVM, PCA et Feature Engineering

Yasmine EL-Ainous

Projet de 3A – Polytech Nice Sophia

## Table des matières

<b>1</b>	<b>Introduction et Exploration des Données</b>	<b>2</b>
<b>2</b>	<b>Pré-traitement et Ingénierie de Caractéristiques</b>	<b>2</b>
<b>3</b>	<b>Expérimentation et optimisation du modèle SVM</b>	<b>4</b>
3.1	Choix du classifieur : Support Vector Machine . . . . .	4
3.2	Construction du pipeline de classification . . . . .	4
3.3	Optimisation des hyperparamètres . . . . .	4
3.4	Évaluation sur le jeu de test . . . . .	4
3.5	Influence du nombre de folds $K$ . . . . .	5
3.6	Stratégies de classification multi-classes . . . . .	5
<b>4</b>	<b>Réseaux de Neurones pour la Classification</b>	<b>5</b>
<b>5</b>	<b>Conclusion</b>	<b>6</b>

# 1 Introduction et Exploration des Données

L'objectif de ce projet est de construire un classifieur robuste pour la reconnaissance de chiffres manuscrits. Le jeu de données utilisé est `digits`, fourni par la bibliothèque Scikit-learn. Il se compose de 1797 images en niveaux de gris de dimensions 8x8 pixels, chaque image étant un vecteur de 64 features correspondant à l'intensité des pixels (de 0 à 16). Chaque image est associée à un label, le chiffre qu'elle représente (de 0 à 9).

Une analyse exploratoire a révélé une distribution des classes quasi-uniforme, avec environ 180 échantillons par chiffre (Figure 1). Cette répartition équilibrée est idéale car elle évite les biais de classification et le besoin de techniques de ré-échantillonnage.

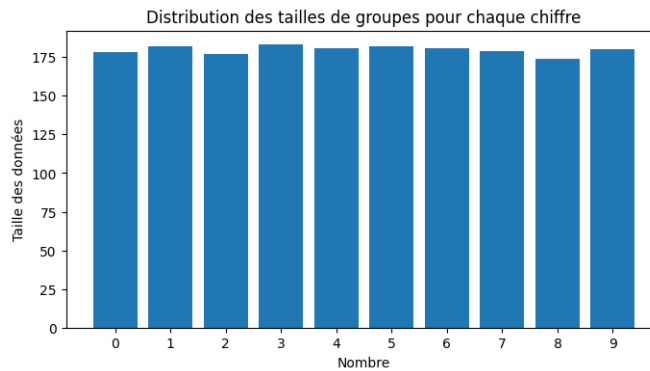


FIGURE 1 – Distribution des classes dans le jeu de données complet. L'échantillon est bien équilibré.

Pour visualiser la structure intrinsèque des données, une Analyse en Composantes Principales (ACP) a été appliquée aux données préalablement normalisées (Min-Max Scaling). Le graphique de la variance expliquée cumulée (Figure 2) montre que plus de 90% de la variance est capturée par les 30 premières composantes principales, justifiant une réduction de dimension. Les projections 3D révèlent que certaines classes (ex : '0', '4') sont naturellement bien séparées, tandis que d'autres présentent des chevauchements ('1' et '7'), soulignant la nécessité de modèles de classification non-linéaires.

## 2 Pré-traitement et Ingénierie de Caractéristiques

Plutôt que d'utiliser directement les 64 pixels bruts, une étape d'ingénierie de caractéristiques (*feature engineering*) a été menée pour construire un vecteur plus riche et compact, guidant ainsi l'apprentissage du modèle. Trois types de caractéristiques ont été extraits et combinés :

- **Caractéristiques via ACP** : Les 30 premières composantes de l'ACP sont conservées. Elles représentent l'information la plus saillante de l'image de manière décorrélée, agissant comme un débruiteur et un compresseur d'information. La reconstruction d'images (Figure 3) valide qu'avec 30 composantes, l'essentiel de la structure visuelle est préservé.

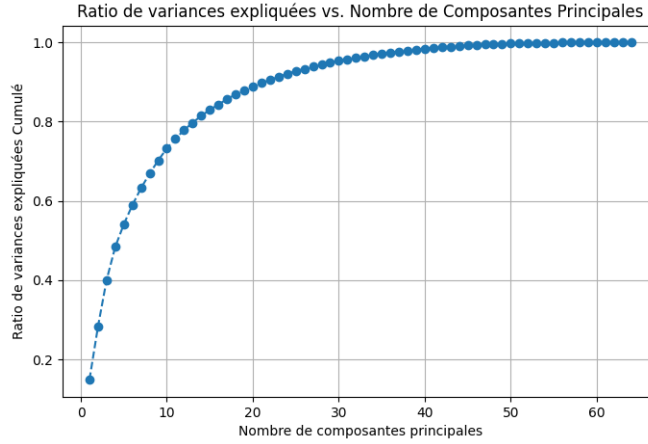


FIGURE 2 – Variance expliquée cumulée en fonction du nombre de composantes principales.

- **Caractéristiques Zonales** : L'image 8x8 est découpée en trois zones horizontales (haut, milieu, bas). La moyenne d'intensité de chaque zone est calculée, résultant en un vecteur de 3 caractéristiques. Cette technique encode une information spatiale grossière, discriminante pour des chiffres comme le '7' (zone haute vide) ou le '1' (zones haute et basse vides).
- **Caractéristiques de Contours (Filtre de Sobel)** : Le filtre de Sobel est appliqué pour détecter les gradients d'intensité, qui correspondent aux contours du chiffre. La moyenne de l'image de Sobel résultante est utilisée comme une seule caractéristique, mesurant la "complexité" ou la "densité" du tracé.

L'ensemble de ces processus a été encapsulé dans un Pipeline Scikit-learn, utilisant une `FeatureUnion` pour l'extraction parallèle des caractéristiques et un `StandardScaler` pour la normalisation finale. Cette approche garantit la reproductibilité et prévient les fuites de données du jeu de test vers l'entraînement. Les données ont été divisées en un ensemble d'entraînement (80%) et de test (20%), un ratio assurant un apprentissage robuste et une évaluation fiable.

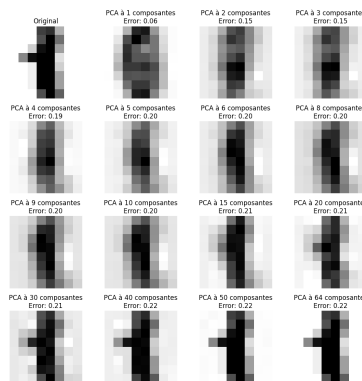


FIGURE 3 – Qualité de reconstruction d'une image en fonction du nombre de composantes ACP.

## 3 Expérimentation et optimisation du modèle SVM

### 3.1 Choix du classifieur : Support Vector Machine

Le Support Vector Machine (SVM) est un classifieur à marge maximale, particulièrement efficace pour des problèmes non linéaires lorsque couplé à un noyau. Dans notre cas, nous utilisons un noyau gaussien (RBF), permettant de projeter les données dans un espace de Hilbert de grande dimension afin de rendre les classes linéairement séparables.

### 3.2 Construction du pipeline de classification

Nous mettons en place une chaîne de traitement (pipeline) combinant :

- Une réduction de dimension via l'Analyse en Composantes Principales (ACP), qui permet de projeter les données sur les vecteurs propres associés à la plus grande variance. Cela a deux intérêts : la débruitisation et la réduction de la complexité.
- Des descripteurs d'intensité issus de la décomposition zonale : on découpe l'image en sous-blocs et on calcule la moyenne d'intensité par zone. Cette approche encode l'information spatiale grossière.
- Un calcul des gradients par le filtre de Sobel, qui capte l'information de texture et de bordure, cruciale pour la classification visuelle.

Ces trois types de caractéristiques sont concaténés pour former une représentation riche mais compacte de l'image.

### 3.3 Optimisation des hyperparamètres

Le SVM avec noyau RBF dépend de deux hyperparamètres essentiels :

- $C$  : pénalité des erreurs de classification. Un  $C$  élevé minimise l'erreur sur l'entraînement mais augmente le risque de surapprentissage.
- $\gamma$  : paramètre du noyau RBF qui contrôle l'influence d'un point d'entraînement. Un  $\gamma$  trop grand conduit à un modèle très complexe et peu généralisable.

Nous effectuons une recherche exhaustive (*GridSearch*) avec validation croisée pour sélectionner les meilleures valeurs de  $C$ ,  $\gamma$  et du nombre de composantes principales  $k$  de l'ACP. Le critère d'évaluation utilisé est l'accuracy moyenne en validation croisée (5 folds).

**Résultats :** Les meilleurs hyperparamètres obtenus sont :

$$C = 10, \gamma = 1, k = 30$$

Accuracy moyenne en cross-validation : 99.1%

### 3.4 Évaluation sur le jeu de test

Le modèle entraîné avec les meilleurs paramètres est évalué sur un jeu de test indépendant. Les résultats sont les suivants :

- **Accuracy** : 98.6%
- **Erreur quadratique moyenne (MSE)** : 0.172

### 3.5 Influence du nombre de folds $K$

Nous avons étudié l'impact du choix de  $K$  dans la validation croisée. Des valeurs faibles de  $K$  (e.g. 2) donnent des estimations moins stables. Au-delà de  $K = 5$ , les gains deviennent marginaux.

### 3.6 Stratégies de classification multi-classes

Les SVM étant conçus pour la classification binaire, deux stratégies d'extension au multi-classes sont employées :

- **One-vs-One (OvO)** :  $\binom{C}{2}$  classifieurs, chacun distinguant une paire de classes. Complexité quadratique en nombre de classes, mais bonne précision.
- **One-vs-Rest (OvR)** :  $C$  classifieurs binaires, chaque classifieur distinguant une classe cible du reste.

Dans notre cas, OvO a fourni une accuracy équivalente (98.6%) à celle d'OvR, avec un temps de calcul légèrement plus élevé. Cela montre que les deux approches sont viables, le choix dépendant du compromis précision/complexité.

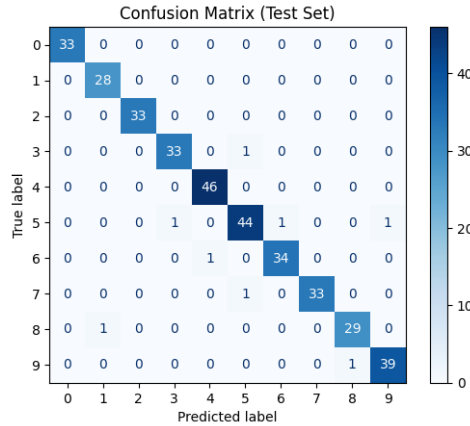


FIGURE 4 – Matrice de confusion du modèle SVM optimisé sur le jeu de test.

## 4 Réseaux de Neurones pour la Classification

En parallèle de l'approche SVM, nous avons exploré un modèle de réseau de neurones (NN) entièrement connecté, utilisant Keras. L'avantage principal de cette approche est sa capacité à apprendre des représentations hiérarchiques directement à partir des pixels bruts, sans ingénierie de caractéristiques manuelle.

Une recherche d'hyperparamètres a été menée sur l'architecture du réseau (nombre de couches, nombre de neurones), la fonction d'activation et l'optimiseur. La meilleure configuration trouvée est un réseau dense avec :

- Une couche d'entrée de 64 neurones (un par pixel).
- **Trois couches cachées de 128 neurones** chacune, avec la fonction d'activation ReLU.

- Une couche de sortie de 10 neurones avec une activation **Softmax**.
- L'optimiseur **RMSprop** et la fonction de perte **SparseCategoricalCrossentropy**.

Ce modèle, entraîné sur 50 époques, atteint une **accuracy de 98.9%** sur le jeu de test. Les courbes d'apprentissage (Figure 5) montrent une convergence rapide et stable, sans signe de sur-apprentissage majeur, ce qui valide la robustesse du modèle.

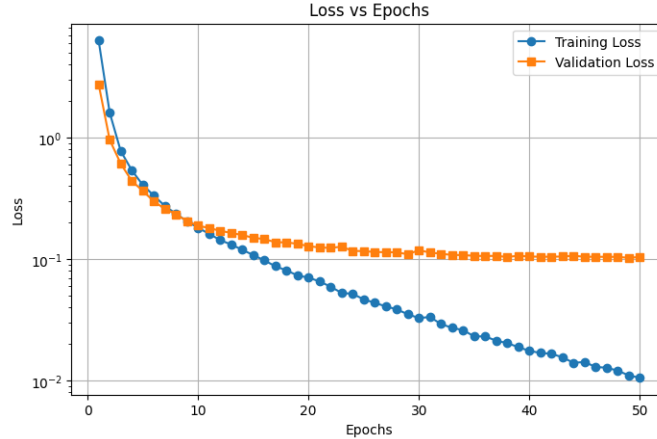


FIGURE 5 – Évolution de la fonction de perte durant l'entraînement du réseau de neurones.

## 5 Conclusion

Ce projet a permis de mettre en œuvre et de comparer deux approches performantes pour la classification d'images. La première, basée sur un SVM avec des caractéristiques conçues manuellement, a démontré qu'une bonne compréhension des données et une ingénierie de caractéristiques pertinente peuvent mener à d'excellents résultats. La seconde, s'appuyant sur un réseau de neurones, a confirmé la puissance des méthodes d'apprentissage profond pour extraire automatiquement des motifs complexes, surpassant légèrement l'approche classique.

TABLE 1 – Synthèse des performances des modèles finaux sur le jeu de test.

Modèle	Ingénierie de Caractéristiques	Accuracy Test
SVM + Noyau RBF	Manuelle (ACP, Zonale, Sobel)	98.6%
Réseau de Neurones Dense	Automatique (Apprentissage profond)	<b>98.9%</b>

Le réseau de neurones a été retenu comme modèle final en raison de sa performance supérieure et de son approche "de bout en bout" plus généralisable.

Les limites de notre étude résident dans la relative simplicité du jeu de données. Pour des tâches de vision par ordinateur plus complexes, l'ingénierie manuelle de caractéristiques deviendrait prohibitive. L'amélioration la plus évidente serait d'explorer les **Réseaux de Neurones Convolutifs (CNN)**, qui sont l'état de l'art en traitement d'images grâce à leur capacité à exploiter la localité spatiale des pixels, et qui offriraient sans doute des performances encore meilleures.