



## « Machine Learning »

### Rapport de projet

# Prédiction de groupe sanguin à partir d'empreintes digitales



## Table de Matière :

1. Introduction .....	3
2. Problématique et objectifs du projet .....	3
3. Description du dataset .....	4
4. Préparation des données .....	4
4.1 Lecture des données .....	4
4.2 Redimensionnement .....	5
4.3 Normalisation .....	6
5. Extraction des caractéristiques .....	7
5.1 Histogramme des gradients orientés (HOG)) .....	7
5.2 Modèle binaire local (LBP).....	8
5.3 Analyse statistique.....	8
7. Modèles de classement .....	9
8. Évaluation des performances .....	12
9. Interface de prédiction .....	14
10. Annexe .....	15
10. Biographie .....	16

## 1. Introduction

L'identification du groupe sanguin est une information vitale dans de nombreux domaines, notamment la médecine d'urgence, les dons de sang, ou encore certaines enquêtes médico-légales. Traditionnellement, cette identification repose sur des analyses biologiques en laboratoire. Cependant, avec l'avancement des technologies d'intelligence artificielle et de traitement d'images, de nouvelles approches innovantes sont explorées pour automatiser ce processus.

Dans le cadre de ce projet, nous nous sommes intéressées à une problématique originale : la **prédiction du groupe sanguin à partir des empreintes digitales**. L'idée est de tirer parti des particularités biométriques des empreintes et de les relier à des caractéristiques biologiques comme le groupe sanguin. Grâce à des techniques de **Machine Learning**, de **vision par ordinateur** et d'**extraction de caractéristiques**, notre objectif est de mettre en place un modèle capable de reconnaître le groupe sanguin à partir d'une simple image d'empreinte digitale.

## 2. Problématique et Objectifs du projet

### 2.1 Problématique

La détermination du groupe sanguin est une tâche indispensable dans le domaine médical, notamment pour les transfusions sanguines, les greffes d'organes et les interventions chirurgicales. Cependant, cette procédure nécessite des tests biologiques qui peuvent être coûteux, longs et parfois peu accessibles. De plus, l'identification du groupe sanguin reste un domaine où le manque de données et les défis de précision des modèles prédictifs sont importants. La principale difficulté réside dans le fait que les données utilisées pour prédire le groupe sanguin sont souvent limitées et difficilement accessibles. Dans ce contexte, l'utilisation de nouvelles sources de données, comme les empreintes digitales, pourrait offrir une solution innovante. Toutefois, la question de la fiabilité et de la précision des modèles prédictifs, particulièrement avec des données limitées, constitue un défi majeur à surmonter.

### 2.2 Objectifs du projet

L'objectif principal de ce projet est de créer un modèle de prédiction du groupe sanguin en utilisant des empreintes digitales. Pour cela, nous utilisons différentes techniques de machine Learning, telles que **SVM**, **forêts aléatoires**, et **KNN**, afin d'analyser les caractéristiques extraites des empreintes et de déterminer le groupe sanguin associé. Le projet vise également à déployer ce modèle à travers une interface Python conviviale.

## 3. Description du dataset



Cette lecture en niveaux de gris simplifie les données d'entrée, car elle réduit la dimensionnalité des images tout en préservant les informations essentielles pour la détection des caractéristiques. En effet, une image couleur (RVB) comporte trois canaux (rouge, vert et bleu), ce qui entraîne un nombre considérablement plus élevé de pixels à traiter. En convertissant l'image en niveaux de gris, nous réduisons cette complexité en conservant uniquement une information monocanale, ce qui facilite l'analyse tout en diminuant le temps de traitement.



## 4.2 Redimensionnement

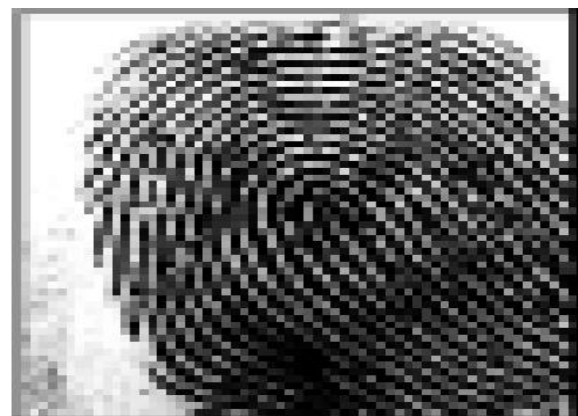
Le redimensionnement des images est une étape essentielle dans notre processus de prétraitement des données. Étant donné que les images d'empreintes digitales proviennent de sources variées, elles peuvent avoir des dimensions différentes. Pour permettre à notre modèle de les traiter de manière cohérente, il est nécessaire de redimensionner toutes les images à une taille uniforme.

Dans notre cas, nous avons choisi de redimensionner chaque image d'empreinte digitale à une taille de 64x64 pixels. Ce redimensionnement simplifie l'analyse tout en conservant les informations cruciales des empreintes. En standardisant la taille des images, nous assurons que chaque entrée pour le modèle est de même forme, ce qui permet au modèle de mieux apprendre les caractéristiques pertinentes des empreintes digitales, comme les motifs ou les structures fines, indépendamment de la résolution initiale des images.

```
# Redimensionner l'image
```

```
img_resized = cv2.resize(img, target_size=(64, 64))
```

Cette étape est essentielle pour uniformiser les images et garantir des dimensions cohérentes, facilitant ainsi l'apprentissage du modèle. Le redimensionnement aide à éviter les variations de taille et de qualité, ce qui permet au modèle d'extraire plus efficacement les caractéristiques discriminantes pour la prédiction du groupe sanguin, tout en optimisant la vitesse d'apprentissage.







statistiques des empreintes digitales, afin d'améliorer la précision de notre modèle de prédiction du groupe sanguin.

```
# 1. Caractéristiques HOG (Histogram of Oriented Gradients)
# Réduire la complexité du HOG
hog_features = hog(image, orientations=8, pixels_per_cell=(16, 16),
                   cells_per_block=(2, 2), visualize=False)
features.extend(hog_features)
```

### 5.1 Histogramme des gradients orientés (HOG))

L'histogramme des gradients orientés (HOG) est une méthode largement utilisée en vision par ordinateur pour la détection d'objets. Elle repose sur l'idée que l'apparence et la forme d'un objet peuvent être bien décrites par la distribution des intensités de gradient ou des directions de contour dans l'image.

Pour appliquer cette méthode, chaque image est d'abord divisée en petites cellules. À l'intérieur de chaque cellule, les gradients de pixels sont calculés, puis regroupés sous forme d'un histogramme basé sur la direction des gradients. Ces histogrammes sont ensuite normalisés pour compenser les variations d'éclairage et de contraste, puis concaténés pour former un vecteur de caractéristiques final.

Dans notre projet, HOG permet d'extraire les structures directionnelles présentes dans les empreintes digitales, comme les lignes de crête et de vallée. Ces structures sont potentiellement discriminantes entre les différents groupes sanguins. L'utilisation de HOG a considérablement contribué à la qualité des représentations utilisées par nos modèles d'apprentissage automatique.

## 5.2 Modèle binaire local (LBP)

Le modèle binaire local (LBP) est une méthode simple mais puissante pour l'analyse de texture. Elle fonctionne en comparant chaque pixel d'une image à ses voisins directs dans un rayon défini. Si un voisin a une valeur de pixel supérieure ou égale à celle du pixel central, on lui attribue un 1, sinon un 0. Cela génère une séquence binaire qui est ensuite convertie en un entier : le code LBP du pixel.

Cette technique permet de résumer les motifs de texture dans une petite région de l'image tout en étant robuste aux variations d'illumination. Dans le contexte des empreintes digitales, LBP est particulièrement efficace pour détecter les micro-textures et les motifs locaux, qui peuvent varier selon la personne et, hypothétiquement, selon le groupe sanguin.

Pour chaque image, nous avons généré une carte LBP, puis extrait un histogramme représentant la fréquence de chaque motif LBP. Cet histogramme est utilisé comme vecteur de caractéristiques pour les modèles de classification.



```
# 2. Local Binary Patterns
# Réduire la complexité du LBP
radius = 2
n_points = 8 * radius
lbp = local_binary_pattern(image, n_points, radius, method='uniform')
hist_lbp, _ = np.histogram(lbp.ravel(), bins=np.arange(0, n_points + 3),
                           range=(0, n_points + 2))
hist_lbp = hist_lbp.astype("float")
hist_lbp /= (hist_lbp.sum() + 1e-7)
features.extend(hist_lbp)
```

### 5.3 Analyse statistique

L'analyse statistique des images constitue une méthode simple mais puissante pour extraire des caractéristiques globales qui peuvent aider à la classification. Dans notre projet, nous avons extrait cinq mesures statistiques fondamentales pour chaque image d'empreinte digitale, en nous basant sur les valeurs des pixels :

- **La moyenne** : elle représente l'intensité lumineuse moyenne de l'image. Elle peut donner une idée générale de la clarté de l'empreinte.
- **L'écart-type** : il mesure la variation ou la dispersion des valeurs de pixel par rapport à la moyenne. Un écart-type élevé suggère une image avec de forts contrastes.
- **La médiane** : elle reflète la valeur centrale de l'intensité lumineuse, ce qui peut être utile pour atténuer l'effet des valeurs extrêmes ou aberrantes.
- **Le maximum** : il indique la valeur de pixel la plus élevée dans l'image, ce qui peut correspondre à des zones fortement marquées de l'empreinte.
- **Le minimum** : il représente la valeur de pixel la plus faible, utile pour détecter les zones les plus sombres de l'empreinte.

Ces cinq caractéristiques sont ajoutées au vecteur de caractéristiques final de chaque image, en complément des descripteurs HOG et LBP. Bien qu'elles ne soient pas directement liées aux structures locales complexes de l'empreinte digitale, ces statistiques globales peuvent enrichir la représentation de l'image en fournissant un contexte général sur son intensité et sa variabilité.

```
# 3. Statistiques basiques de l'image
features.extend([
    np.mean(image),      # Moyenne
    np.std(image),       # Écart-type
    np.median(image),    # Médiane
    np.max(image),       # Maximum
    np.min(image)        # Minimum
])
```

## 7. Modèles de classement







```
def train_and_evaluate_models(X_train, y_train, X_val, y_val, X_test, y_test, classes):

    models_config = {
        'SVM': {
            'model': SVC(probability=True, random_state=42),
            'params': {
                'C': [0.1, 1],
                'kernel': ['rbf', 'linear']
            }
        },
        'Random Forest': {
            'model': RandomForestClassifier(random_state=42, n_jobs=-1),
            'params': {
                'n_estimators': [50, 100],
                'max_depth': [None, 10]
            }
        },
        'KNN': {
            'model': KNeighborsClassifier(n_jobs=-1),
            'params': {
                'n_neighbors': [3, 5],
                'weights': ['uniform']
            }
        }
    }
```

```
results = {}

# Entraînement et évaluation de chaque modèle
for name, config in models_config.items():
    logging.info(f"\nEntraînement du modèle {name}...")

    # Optimisation des hyperparamètres
    logging.info("Optimisation des hyperparamètres...")
    optimization_results = optimize_hyperparameters(
        config['model'], config['params'], X_train, y_train
    )

    # Configuration du modèle avec les meilleurs paramètres
    model = config['model'].set_params(**optimization_results['best_params'])

    # Validation croisée
    cv_scores = cross_val_score(model, X_train, y_train, cv=5, n_jobs=-1)

    # Entraînement final
    model.fit(X_train, y_train)
```

```
# Prédiction
y_pred_val = model.predict(X_val)
y_pred_test = model.predict(X_test)

# Évaluation
val_report = classification_report(y_val, y_pred_val, target_names=classes, output_dict=True)
test_report = classification_report(y_test, y_pred_test, target_names=classes, output_dict=True)

# Matrice de confusion
cm = confusion_matrix(y_test, y_pred_test)

# Sauvegarde des résultats
results[name] = {
    'model': model,
    'val_accuracy': val_report['accuracy'],
    'test_accuracy': test_report['accuracy'],
    'val_report': val_report,
    'test_report': test_report,
    'confusion_matrix': cm,
    'cv_scores': cv_scores.tolist(),
    'best_params': optimization_results['best_params'],
    'optimization_score': optimization_results['best_score']
}
```



```
results = train_and_evaluate_models(  
    X_train_scaled, data['y_train'],  
    X_val_scaled, data['y_val'],  
    X_test_scaled, data['y_test'],  
    data['classes']  
)  
  
output_dir = 'models'  
for model_name, result in results.items():  
    plot_confusion_matrix(  
        result['confusion_matrix'],  
        data['classes'],  
        model_name,  
        output_dir  
    )  
  
# Sauvegarde des résultats  
save_results(results, scaler, output_dir)
```

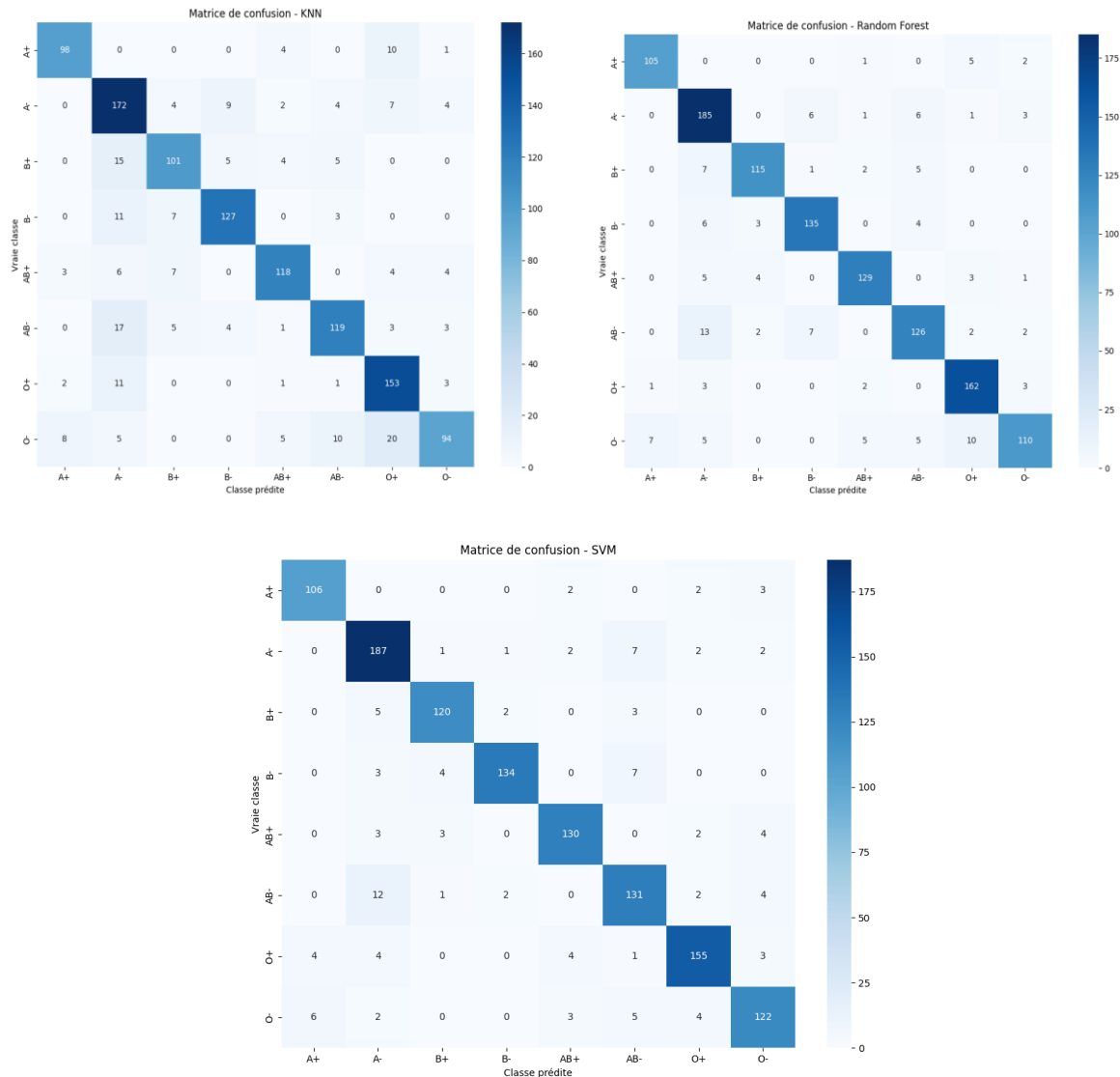
Ce code entraîne et compare trois modèles (SVM, Random Forest, KNN) pour prédire le groupe sanguin à partir d'empreintes digitales. Il évalue leurs performances via des matrices de confusion et des métriques de précision, puis sauvegarde le meilleur modèle et ses résultats dans le dossier `models/` pour une utilisation ultérieure.

## 8. Évaluation des performances

### 8.1. Résultats d'optimisation des paramètres :

Modèle	Meilleurs Paramètres	Précision(Validation)	Précision(Test)	Score CV moyen
SVM	{'kernel': 'linear', 'C': 0.1}	90.08%	90.42%	0.8947 (±0.0048)
RF	{'n_estimators': 100, 'max_depth': None}	88.67%	88.92%	0.8733 (±0.0051)
KNN	{'weights': 'uniform', 'n_neighbors': 5}	83.50%	81.83%	0.8172 (±0.0128)

Les résultats montrent que le SVM avec un noyau linéaire ( $C=0.1$ ) obtient la meilleure performance (90.08% en validation, 90.42% en test), suivi de près par le Random Forest (88.92% en test). Le KNN se révèle moins précis (81.83%), probablement en raison de sa sensibilité aux variations des données. Les scores de validation croisée (CV) confirment cette hiérarchie, avec une stabilité notable pour le SVM ( $\pm 0.48\%$  d'écart-type).



Les matrices de confusion confirment les bonnes performances globales des trois modèles, avec toutefois des nuances importantes. Le SVM se distingue particulièrement par sa capacité à correctement classifier la majorité des groupes sanguins, comme en témoignent les valeurs élevées sur la diagonale principale de sa matrice (ex: 106 vraies prédictions pour A+, 134 pour B+). Bien que le Random Forest et le KNN montrent également des résultats satisfaisants.

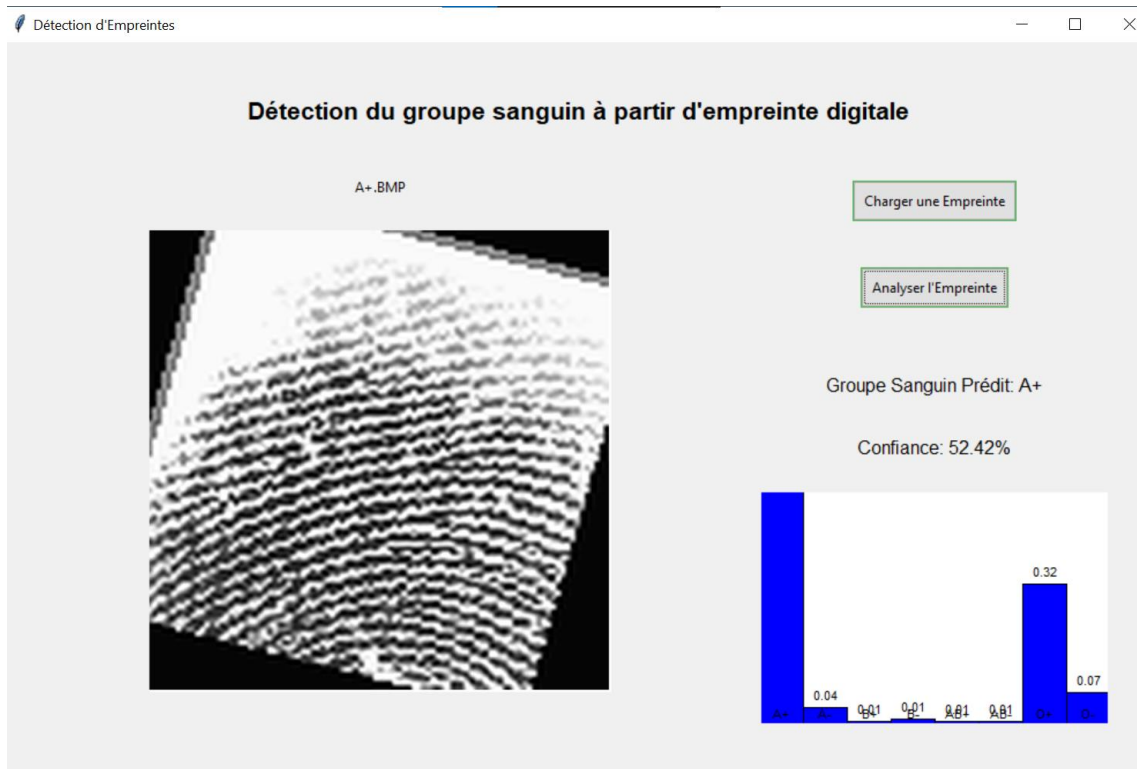
Le SVM combine plusieurs avantages décisifs :

1. **Précision supérieure** (90.42% contre 88.92% et 81.83%)
2. **Stabilité des prédictions** (faible écart validation/test)
3. **Robustesse aux variations** (meilleure gestion des cas ambigus comme AB- et O-)

Cette combinaison de performance, de fiabilité et de simplicité (noyau linéaire) fait du SVM le choix optimal pour cette application, malgré un temps de calcul légèrement plus long.

Sa capacité à maintenir une haute précision tout en minimisant les erreurs critiques (comme les confusions entre groupes rares) justifie amplement sa sélection pour le déploiement final

## 9. Interface de prédiction



Cette interface permet de prédire le groupe sanguin d'un individu à partir de l'analyse de son empreinte digitale.

### Fonctionnalités principales :

1. Chargement de l’empreinte : L'utilisateur peut importer une image d'empreinte digitale via le bouton "*Charger une Empreinte*".
2. Analyse automatisée : Le système traite l’empreinte via le bouton "*Analyser l'Empreinte*" pour en extraire des caractéristiques biométriques.
3. Résultats de prédiction :
  - Groupe sanguin prédit : Affiche le résultat (ex: "A+").
  - Niveau de confiance : Indiqué en pourcentage (ici 52.42%), reflétant la fiabilité de la prédiction.
  - Probabilité d’appartenance à chaque groupe sanguin (graphe)





## 10. Annexe



MachineLearningPres  
entation.pptx



## 10. Webographie

Titre	Date	Lien
<b>Blood Group Detection Using Fingerprint Images</b>	28/04/2025	<a href="https://www.ijraset.com/research-paper/blood-group-detection-using-fingerprint-images">https://www.ijraset.com/research-paper/blood-group-detection-using-fingerprint-images</a>
<b>A Review of Fingerprint Image Pre-processing</b>	28/04/2025	<a href="https://www.researchgate.net/publication/270975624_A_Review_of_Fingerprint_Image_Pre-processing">https://www.researchgate.net/publication/270975624_A_Review_of_Fingerprint_Image_Pre-processing</a>
<b>Blood Group Detection Using Fingerprint and Scanner</b>	28/04/2025	<a href="https://ijirt.org/publishedpaper/IJIRT170701_PAPER.pdf">https://ijirt.org/publishedpaper/IJIRT170701_PAPER.pdf</a>
<b>Blood group determination using fingerprint.</b>	29/04/2025	<a href="https://www.matec-conferences.org/articles/mateconf/pdf/2024/04/mateconf_icmed2024_01069.pdf">https://www.matec-conferences.org/articles/mateconf/pdf/2024/04/mateconf_icmed2024_01069.pdf</a>
<b>Etude d'un système complet de reconnaissance d'empreintes digitales pour un capteur microsysteme à balayage</b>	29/04/2025	<a href="https://theses.hal.science/file/index/docid/48244/filename/tel-00009742.pdf">https://theses.hal.science/file/index/docid/48244/filename/tel-00009742.pdf</a>