République Tunisienne
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique
Université de Carthage

Institut Nationale des Sciences Appliquées et de Technologie

Rapport de Projet de Fin d'année

4ème Année

Réseaux Informatiques et Télécommunications

Conception d'un dispositif IoT et application mobile pour une évaluation de la qualité de l'eau basée sur le Machine Learning

Réalisé par:

Sofiene Chaabouni

Lobna Sellami

Eya Soussi

Encadrante:

Dr. Rabaa Youssef

Examinatrice:

Dr. Hamdi Sana

Année universitaire 2022 / 2023

Remerciements

Nous tenons à exprimer notre profonde gratitude envers notre encadrante, Dr. Rabaa YOUSSEF. C'est grâce à sa guidance et à son soutien que nous avons pu réaliser ce projet, qui a été à la fois nouveau pour nous et extrêmement enrichissant. Sa confiance en nos capacités et son expertise nous ont permis de repousser nos limites et d'atteindre des résultats dont nous sommes fiers. Nous sommes reconnaissants d'avoir eu l'opportunité de travailler sous sa direction, et nous lui sommes très reconnaissants pour l'impact positif qu'elle a eu sur notre parcours académique et professionnel.

Table des matières

Introduction générale

Notre projet est dédié au développement d'une solution novatrice pour l'évaluation de la turbidité de l'eau, en combinant des technologies telles que la capture d'images, l'Internet des objets (IoT) et l'apprentissage automatique (Machine Learning). L'objectif principal de notre projet est de fournir une méthode plus rapide, plus accessible et plus précise pour mesurer la qualité de l'eau, en éliminant les contraintes et les limites des méthodes traditionnelles.

La turbidité de l'eau est un indicateur essentiel de sa qualité, car elle mesure la quantité de particules en suspension dans l'eau. Les méthodes conventionnelles de mesure de la turbidité impliquent souvent l'utilisation d'équipements coûteux et complexes, ainsi que des procédures laborieuses. Notre projet vise à simplifier ce processus en utilisant une approche basée sur l'imagerie et l'analyse des images capturées à l'aide d'une application mobile.

Notre solution repose sur une architecture composée d'une application mobile, d'un appareil IoT et d'un modèle d'apprentissage automatique. L'application mobile permet aux utilisateurs de capturer des images des échantillons d'eau, qui sont ensuite traitées et analysées par un modèle de prédiction de la turbidité.

L'aboutissement de notre projet serait une solution pratique, précise et accessible pour l'évaluation de la turbidité de l'eau, ouvrant ainsi de nouvelles perspectives dans le domaine de la surveillance de la qualité de l'eau. Notre solution pourrait être utilisée dans divers contextes, tels que la surveillance des eaux de surface, l'évaluation de la qualité de l'eau potable, la gestion des ressources hydriques et la protection de l'environnement.

Dans les sections suivantes de notre rapport, nous présenterons en détail les différentes étapes de notre démarche, de la conception et la réalisation du système à l'expérimentation et à l'analyse des résultats. Nous mettrons en évidence les choix technologiques, les spécifications des besoins, les résultats obtenus et les perspectives d'amélioration de notre solution. L'objectif ultime est de contribuer à l'amélioration de la surveillance de la qualité de l'eau grâce à une approche innovante et technologiquement avancée.

Chapitre 1

Contexte général du projet et Spécification des besoins pour la solution

Ce chapitre s'attache à fournir un contexte général et à réaliser une analyse approfondie des besoins. Nous explorerons les défis actuels auxquels sont confrontées les méthodes traditionnelles de mesure de la turbidité, nous soulignerons l'importance d'une évaluation précise et opportune de la qualité de l'eau et nous discuterons des avantages et des implications potentiels de la solution que nous proposons. En comprenant les limites et les exigences existantes dans le domaine, nous pouvons mieux contextualiser l'importance et la pertinence de notre projet d'étude, tout en identifiant l'impact potentiel qu'il peut avoir sur les différents secteurs et parties prenantes impliqués dans la gestion des ressources en eau.

1.1 Contexte général du projet

On va se permettre de situer notre projet dans un cadre plus large et de définir les principaux éléments qui motivent sa réalisation.

1.1.1 Contexte du projet

L'eau est une ressource vitale indispensable à la vie humaine, à la sécurité alimentaire, ainsi qu'à l'industrie. Elle devient de plus en plus chère et valorisée, étant donné que nous faisons face à une sécheresse en Tunisie ainsi que dans plusieurs autres pays du monde. La qualité de l'eau est donc un paramètre essentiel à contrôler pour s'assurer qu'elle soit adéquate pour toute utilisation. L'une des principales mesures de qualité de l'eau est la turbidité, une mesure de la quantité de particules en suspension ou de matières solides présentes dans l'eau. Ces particules peuvent inclure des sédiments, des matières organiques, des minéraux, des algues et d'autres substances.

La turbidité est une caractéristique importante de la qualité de l'eau, car elle affecte directement sa clarté et sa transparence. En pratique, le turbidimètre est largement utilisé comme moyen de détection de la turbidité. Il fonctionne en prélevant des échantillons d'eau sur le site cible, puis en les analysant en laboratoire à l'aide de turbidimètres.

Cependant, cette méthode présente plusieurs inconvénients. Elle peut être coûteuse, nécessitant l'achat et l'entretien d'équipements spécifiques. De plus, elle exige l'expertise de professionnels formés pour réaliser les analyses et interpréter les résultats. En outre,

cette approche ne permet pas une évaluation en temps réel de la turbidité, ce qui peut limiter sa pertinence dans des situations nécessitant une réactivité immédiate.

Face à ces limites et avec l'émergence de l'intelligence artificielle, on observe un développement de méthodes de détection de la turbidité basées sur la vision par ordinateur. Ces méthodes, qui constituent une alternative ou un complément aux méthodes traditionnelles, se fondent sur l'apprentissage machine et l'apprentissage profond.

C'est dans ce contexte que notre projet intervient, en proposant une solution innovante grâce à une application mobile conçue pour la collecte d'images et une plateforme pour l'enregistrement en temps réel des valeurs de turbidité. Ces données seront ensuite utilisées pour former des modèles d'apprentissage machine pour la tâche de classification.

Ce rapport synthétise ce qui a été réalisé, en commençant par une introduction du contexte général. La deuxième partie est consacrée aux spécifications techniques du projet. Dans les parties suivantes, nous présenterons la conception du projet et les réalisations obtenues. Enfin, nous présenterons les expérimentations et résultats obtenus.

1.1.2 Autres travaux et Axes d'amélioration proposés

Plusieurs réalisations ont abordé le sujet de l'évaluation de la qualité de l'eau en utilisant des méthodes de vision par ordinateur. En effet, pour des sites à grande échelle comme les lacs, les rivières et les océans, des images satellites ont été utilisées pour estimer à distance la qualité de l'eau. L'avantage de cette technique est qu'elle élimine la nécessité d'effectuer des prélèvements sur place et permet de couvrir d'immenses zones géographiques. Cependant, la fréquence des survols des satellites et le coût élevé des images satellites rendent cette méthode peu pratique pour une analyse continue [1].

D'autres projets se sont basés sur des jeux de données contenant des images de verres d'eau[2]. Cependant, leurs modèles restent imprécis, étant donné que l'eau, objet transparent, est pauvre en caractéristiques et en informations utiles, ce qui complique la tâche d'extraction des fonctionnalités. De plus, les résultats sont généralement sensibles aux conditions d'acquisition, telles que le contraste, la luminosité et les tâches aberrantes de réflexion. La résolution de la caméra peut également influencer le modèle, puisque les images à haute résolution offrent davantage de détails.

L'analyse des travaux précédemment réalisés a révélé certains défauts qui ont suscité notre attention. Dans le cadre de notre projet, nous avons donc défini comme objectif principal la création d'une base de données plus stable et la mise en œuvre de techniques de classification pour évaluer à la fois la qualité de cette base de données et les performances des classificateurs.

1.2 Spécification des besoins pour la solution

La spécification des besoins pour la solution est cruciale dans notre étude, car elle vise à définir de manière précise les exigences et les attentes que notre système doit satisfaire. Cette section permet de traduire les objectifs et les contraintes identifiés dans le contexte général en spécifications claires et mesurables.

1.2.1 Application mobile

Dans cette partie on va se concentrer sur le développement et les fonctionnalités de l'interface utilisateur destinée à la capture d'images de turbidité de l'eau.

1.2.1.1 Spécifications conceptuelles et Diagramme de cas d'utilisation

Le besoin d'une estimation rapide et pratique de la qualité d'eau est devenu très important pour la réalisation efficace, peu chère et précise d'un système de contrôle. L'introduction des systèmes utilisant la vision par ordinateur pour l'estimation de la turbidité requiert l'utilisation d'une application mobile pour la collecte et l'intégration de données et même le déploiement ultérieur d'un modèle d'intelligence artificielle.
Notre application a pour objectif principal l'acquisition de données afin de nous permettre, dans un second temps, d'entraîner le modèle de classification. Elle se concentre sur la région d'intérêt, recadre l'image automatiquement , puis la charge vers le cloud avec sa valeur de turbidité récupérée depuis la plateforme IoT.

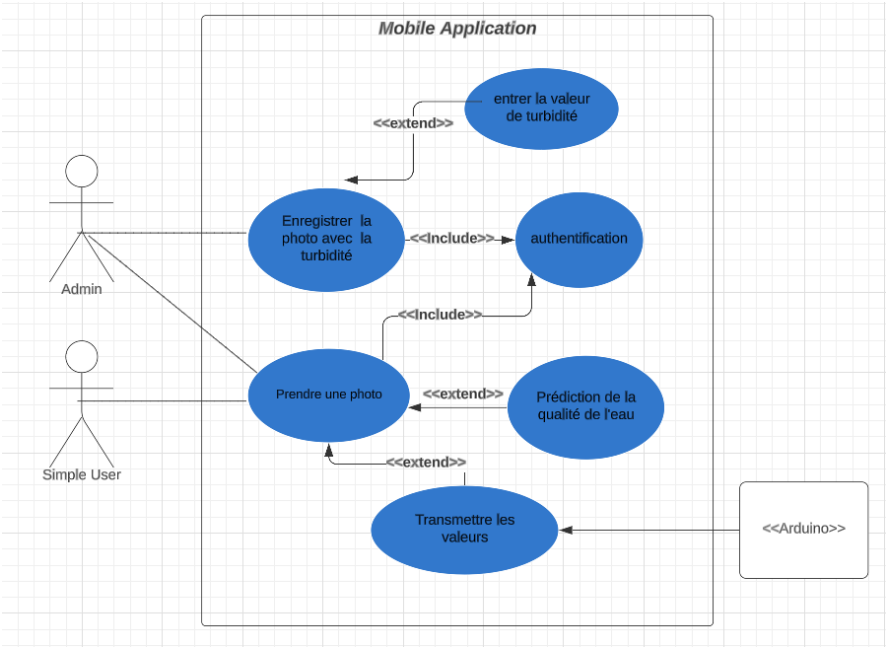


FIGURE 1.1 – Diagramme de cas d'utilisation de l'application Turbidity App.

Comme montré dans la figure 1.1, l'administrateur et l'utilisateur simple de notre application peuvent prendre des photos d'échantillons d'eau avec un motif en arrière-plan. La différence entre ces deux acteurs selon le diagramme de cas d'utilisation est que l'administrateur est capable de construire un dataset à partir des photos qu'il prend, en extrayant la valeur de turbidité soit à partir d'un capteur s'il en utilise un, soit en saisissant cette valeur manuellement s'il se trouve dans un laboratoire équipé d'un turbidimètre.
En revanche, l'utilisateur simple prend simplement des photos dans le but d'utiliser les modèles d'inférence et de prédire la classe correspondante de l'échantillon pris en photo.

Besoins fonctionnels

- Permettre une acquisition facile et pratique des données ainsi que la récupération de la turbidité correspondante(label).
- Permettre une sauvegarde des images dans la base de données avec la valeur de turbidité et la date indiqués dans le nom .
- Sauvegarder les coordonnées de chaque utilisateur dans le cloud.

— Prendre une photo dans le but d'utiliser les modèles d'inférence et de prédire la classe correspondante .

Besoins non fonctionnels

— L'application mobile doit capturer et recadrer les images rapidement et efficacement.
— La communication entre l'application mobile et l'appareil IoT doit avoir une latence minimale.
— La prédiction du modèle doit être effectuée en temps voulu, en fournissant des résultats en temps réel ou quasi réel.
— L'application mobile doit pouvoir gérer un grand nombre d'utilisateurs simultanés et de demandes de traitement d'images.
— L'application mobile doit avoir une interface conviviale, permettant aux utilisateurs de capturer et de recadrer facilement des images d'eau.
— L'application doit être légère en termes d'espaces de stockage et par conséquent le modéle à déployer .

1.2.1.2 Diagramme de séquence

Pour schématiser la vue comportementale de notre application ,nous faisons recours au diagramme de séquence d'UML comme montré dans les figures 1.2 et 1.3 . Ce diagramme permet de précisier les interactions entre l'acteur et le système avec des messages présentés dans un ordre chronologique.

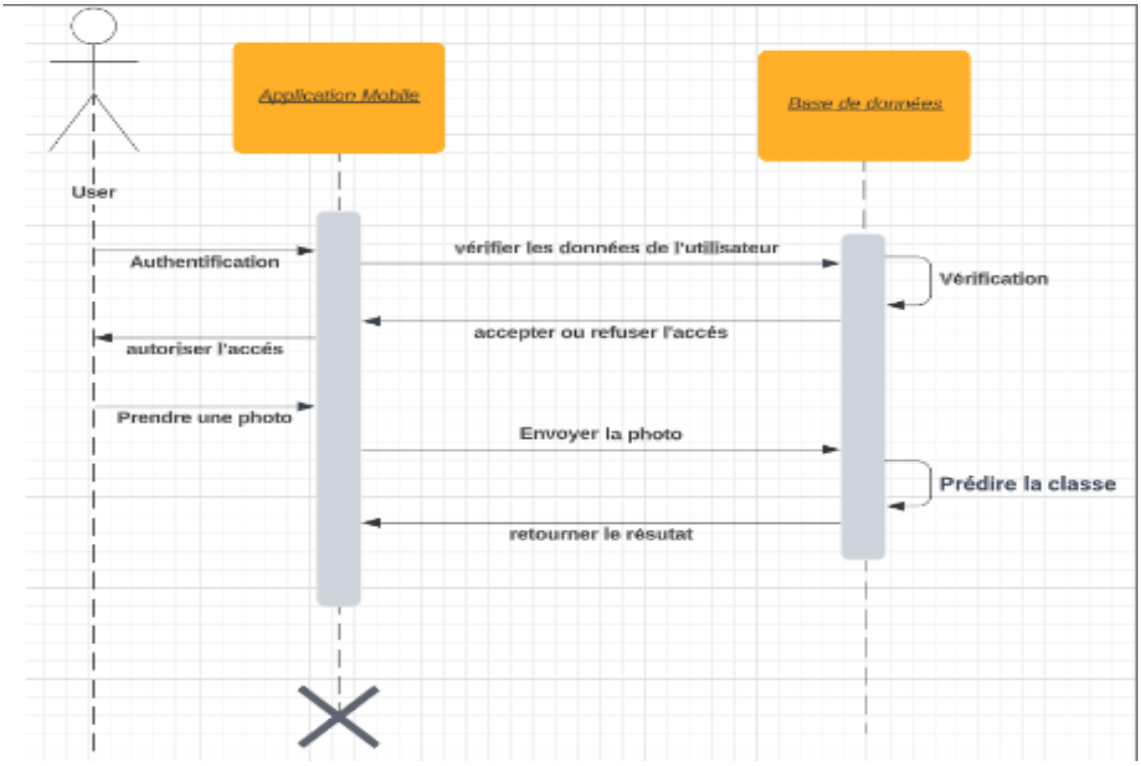


FIGURE 1.2 – Diagramme de séquence pour un simple utilisateur

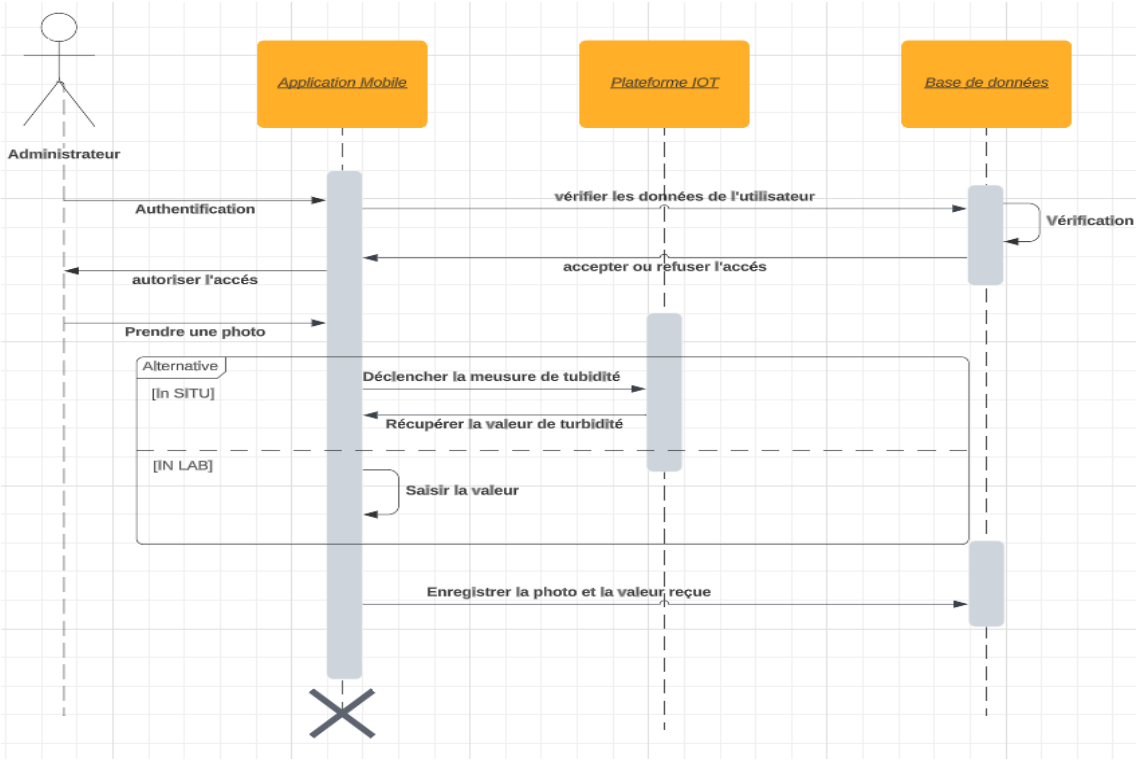


FIGURE 1.3 – Diagramme de séquence pour un admin

Selon les figures 1.2 et 1.3, le processus d'utilisation de l'application implique une étape d'authentification pour l'utilisateur. Une fois authentifié, l'administrateur dispose de différentes options pour capturer une photo et obtenir la valeur de turbidité correspondante. S'il dispose d'un capteur, il peut envoyer une requête à la plateforme IoT pour récupérer la valeur de turbidité. En revanche, s'il est dans un laboratoire, il peut entrer manuellement la valeur de turbidité. Après avoir obtenu la valeur de turbidité, l'administrateur peut sauvegarder la photo avec la valeur correspondante dans la base de données. Cela permet de constituer un dataset qui sera utilisé pour entraîner les modèles et améliorer leur précision. D'autre part, pour un utilisateur simple, le processus consiste à prendre une photo et l'envoyer au serveur Firebase. Le serveur utilise ensuite les modèles d'inférence pour prédire la classe de la photo. Le résultat de la prédiction est renvoyé à l'utilisateur, lui permettant ainsi d'obtenir une estimation de la turbidité de l'échantillon d'eau.

1.2.2 Plateforme IoT

La sélection d'un dispositif IoT adapté est essentielle pour assurer une collecte efficace et fiable des données. Pour répondre à nos besoins spécifiques,nous devons tenir compte des critères fonctionnels et non fonctionnels suivants :

Besoins non fonctionnels :

— Facilité d'utilisation : Le dispositif IoT doit être convivial et intuitif, avec une interface utilisateur facile à comprendre et à utiliser pour nous simplifier la configuration, la gestion et la surveillance.

— Fiabilité : Le dispositif IoT doit être fiable, capable de fonctionner de manière stable et constante, afin de garantir une collecte de données sans interruption ni perte.

En tenant compte de ces besoins fonctionnels et non fonctionnels, nous serons en mesure de sélectionner un dispositif IoT adapté qui répondra à nos exigences spécifiques et contribuera au succès de notre projet.

1.2.3 Computer vision

La partie Computer Vision de notre projet vise à améliorer le protocole d'acquisition existant en adoptant une approche différente par rapport aux travaux précédents. Nous avons constaté que les techniques de recadrage utilisées dans les travaux antérieurs étaient fastidieuses, car ils prenaient d'abord la photo puis recadraient l'image.Dans les figures 1.4, nous présentons quelques images extraites de l'ancien dataset utilisé dans le cadre de ces travaux. Ces images illustrent les différents échantillons de turbidité étudiés Dans notre approche, nous capturons directement une photo déjà recadrée, ce qui nous permet d'obtenir un dataset plus propre et de disposer d'un protocole d'acquisition plus robuste.

Pour garantir des prédictions précises et fiables, nous avons identifié certains besoins fonctionnels et non fonctionnels importants.

FIGURE 1.4 – Quleques images extraites de l'ancien dataset

Besoins fonctionnels

— Notre modèle doit être capable de valider le dataset avec une exactitude (accuracy) de 90 %. Cela signifie que notre modèle doit être capable de classifier correctement au moins 90 % des exemples du dataset, en fournissant des prédictions cohérentes avec les valeurs de turbidité réelles. Cette validation est cruciale pour s'assurer de la qualité des données utilisées dans l'entraînement du modèle.

Besoins non fonctionnels :

— Le modéle doit avoir une exactitude maximale de nos prédictions en minimisant le nombre d'erreurs de classification.

L'objectif final de cette partie est de développer un modèle de classification précis pour estimer la turbidité de l'eau à partir des images capturées par notre application mobile. Ce modèle amélioré nous permettra de fournir en temps réel des informations sur la

qualité de l'eau, contribuant ainsi à une gestion plus efficace des ressources hydriques et à la prise de décisions éclairées.

1.3 Conclusion

En conclusion, le développement d'une application mobile combinée à un dispositif IoT pour l'évaluation de la turbidité de l'eau représente une avancée significative dans le domaine de la surveillance de la qualité de l'eau. Cette solution innovante répond aux limites des méthodes traditionnelles de mesure de la turbidité en s'appuyant sur le traitement d'images, la technologie IoT et l'apprentissage automatique. En combinant leurs puissances, nous avons développé une solution qui simplifie le processus de mesure de la turbidité.

Chapitre 2

Conception et réalisation

Cette section est dédiée à la conception et à la mise en place concrète de notre système, en tenant compte des spécifications et des besoins identifiés précédemment. Nous détaillerons les différentes étapes de notre démarche, de la conception de l'architecture globale à l'implémentation des fonctionnalités spécifiques. Nous mettrons en évidence les choix technologiques effectués et les décisions prises tout au long du processus de développement. Notre objectif est de créer un système fonctionnel et performant, capable de répondre aux besoins et aux exigences définis dans notre projet d'étude.

2.1 Section IoT et Mobile

Dans cette section, nous explorerons les aspects liés à la communication et à la collecte des données à partir des capteurs IoT, ainsi que l'intégration de ces données dans notre application mobile. Nous mettrons en évidence les protocoles de communication utilisés ainsi que les technologies et les frameworks qui facilitent l'échange de données entre les appareils IoT et l'application mobile. Notre objectif est de créer une synergie efficace entre l'application mobile et les dispositifs IoT, permettant une collecte de données en temps réel et une expérience utilisateur fluide et intuitive.

2.1.1 La solution proposée

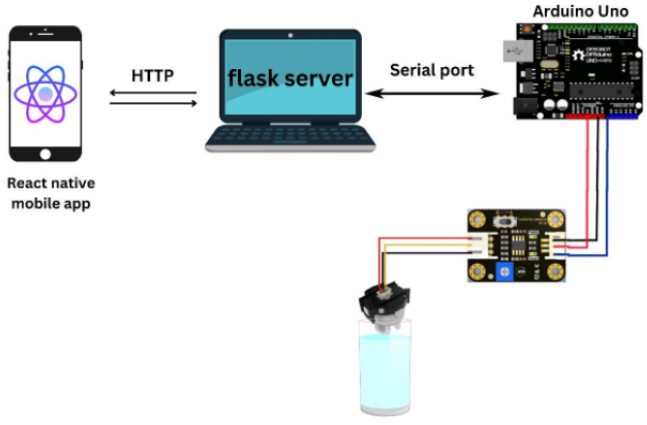


FIGURE 2.1 – Solution proposée

La figure 2.1 donne une vision globale de la solution qu'on propose.En effet, elle repose sur un serveur web Flask qui permet d'écouter les données envoyées par un Arduino uno connecté à un port série (COM4). Les données sont ensuite converties en format JSON et renvoyées en réponse à une requête GET envoyée par une application mobile. Il est important de noter que cette solution est conçue pour fonctionner sur un ordinateur local disposant d'une connexion série disponible sur le port COM4. Pour garantir le bon fonctionnement de l'application, il est également nécessaire de s'assurer que le téléphone et l'ordinateur sont connectés sur le même réseau LAN. Cette solution offre une manière simple et efficace de récupérer les données de l'Arduino et de stocker efficacement ces données en temps réel . Il convient de souligner que c'est au niveau de l'application qu'a lieu le recadrage de l'image, l'attente de la récupération de la valeur NTU, et finalement l'enregistrement de ces données dans la base de données Firebase.

2.1.2 Diagramme de classe

Le diagramme de classes est considéré comme le plus important dans la modélisation orientée objet, et il est le seul obligatoire lors d'une telle modélisation. Le diagramme de classes illustre la structure interne du système. Il offre une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation. Il s'agit d'une vue statique, car on ne prend pas en compte le facteur temporel dans le comportement du système.

Les principaux éléments de cette vue statique sont les classes et leurs relations : l'association, la généralisation, et plusieurs types de dépendances, telles que la réalisation et l'utilisation.

Une classe est une description d'un groupe d'objets partageant un ensemble commun de propriétés (les attributs), de comportements (les opérations ou méthodes) et de relations avec d'autres objets (les associations et les agrégations).Une classe de conception est composée par :

- **Attributs :** chaque attribut d'une classe est le même pour chaque instance de cette classe.
- **Méthodes :** elles définissent le comportement d'une classe elle-même et non le comportement de ses instances, qui peut être différent.

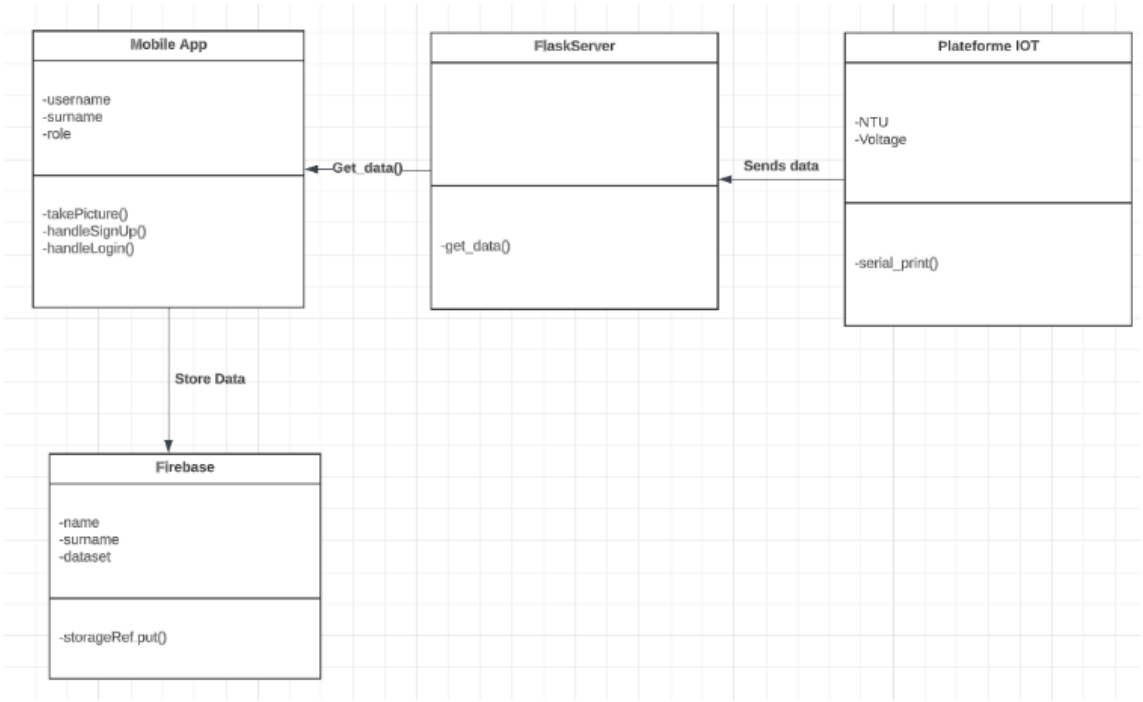


FIGURE 2.2 – Diagramme de Classe

Ce diagramme de classe montré dans la figure 2.2 permet de visualiser l'architecture globale de l'application, ainsi que les interactions entre les différentes parties impliquées y compris l'application mobile, le serveur, la plateforme IoT et le serveur de la base de données. Chacune de ces parties a des rôles spécifiques dans le fonctionnement de l'application. L'application mobile récupère la valeur de turbidité à partir du serveur, qui lui-même la récupère de la plateforme IoT. Le serveur de la base de données est responsable du stockage des images et des données des utilisateurs. L'interaction entre ces parties permet à l'application de fonctionner de manière efficace et fluide.

2.1.3 Diagramme de séquence

Voici le diagramme de séquence de toute l'architecture mise en place qui détaille par ordre chronologique l'interaction entre les différentes composantes du système

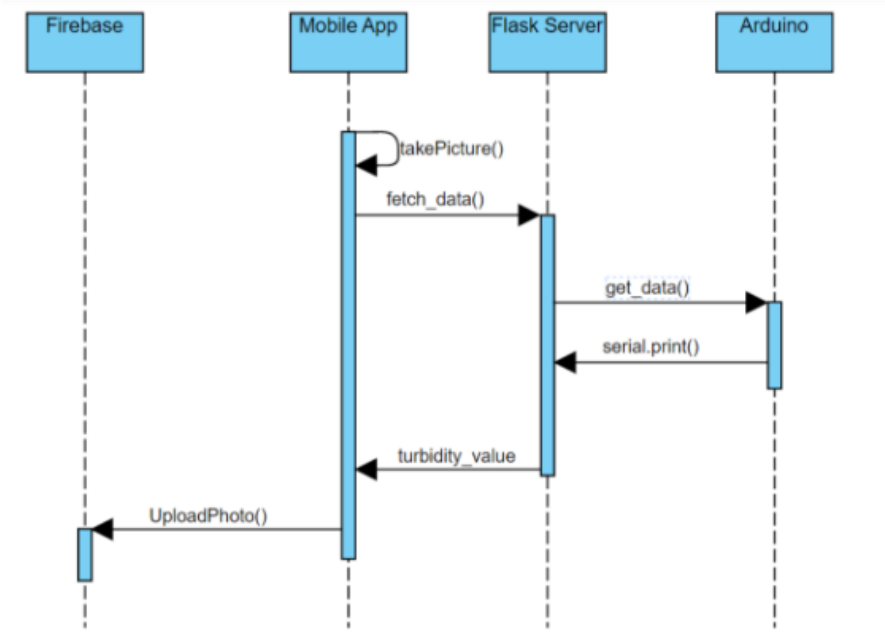


FIGURE 2.3 – Diagramme de Séquence de l'architecture globale

Le diagramme de séquence de la Figure 2.3 représente le processus de construction du dataset tel qu'illustré dans la Figure 2.1. L'utilisateur commence par prendre une photo et envoie une requête pour récupérer la valeur de turbidité au serveur. Le serveur récupère cette valeur à partir du capteur de la carte Arduino. Une fois la valeur retournée, l'application se charge de sauvegarder l'image avec sa valeur correspondante. Ce diagramme permet de visualiser clairement les étapes impliquées dans la construction du dataset.

2.1.4 Technologies utilisées

Dans cette section, nous allons présenter les différents outils, langages de programmation, frameworks et technologies que nous avons choisis pour la conception et la réalisation de notre solution. Cette section met en évidence les choix technologiques pertinents qui ont permis de développer une solution robuste, efficace et adaptée à nos objectifs.

2.1.4.1 Environnement matériel

Capteur de turbidité : TSW-20M

Le capteur de turbidité TSW-20M est un instrument conçu pour évaluer la qualité de l'eau en détectant les niveaux de turbidité. Il fonctionne en utilisant la lumière pour détecter les particules en suspension dans l'eau. La méthode repose sur le principe que lorsque la lumière passe à travers une eau contenant des particules en suspension, une partie de cette lumière est dispersée par ces particules. Le capteur de turbidité mesure alors la quantité de lumière qui est dispersée, ce qui donne une indication sur la quantité de particules en suspension, et donc sur la turbidité de l'eau.

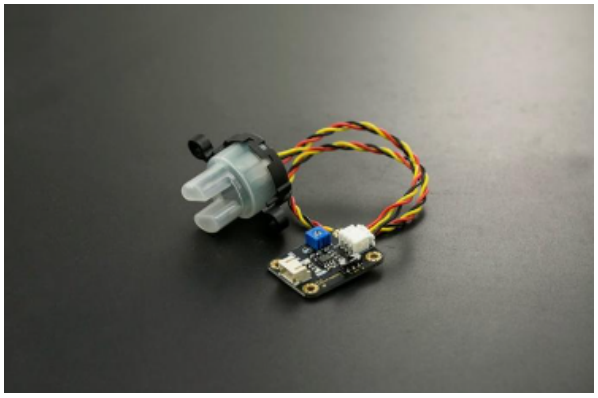


FIGURE 2.4 – Le capteur accompagné du convertisseur analogique-numérique.

Arduino UNO

L'Arduino Uno est une carte microcontrôleur. Il suffit simplement de la connecter à un ordinateur avec un câble USB ou de l'alimenter avec un adaptateur AC-DC. L'Arduino Uno est particulièrement facile à utiliser pour les débutants grâce à son programme d'amorçage intégré, qui permet de charger de nouveaux programmes sur la carte sans nécessiter de matériel de programmateur externe.

Dans le cadre de notre projet, l'Arduino Uno est utilisé pour lire les données du capteur de turbidité, convertir le signal analogique en signal numérique grâce à son convertisseur analogique-numérique intégré, et les transmettre à l'ordinateur qui se chargera en tant que serveur et de les envoyer à l'application mobile .

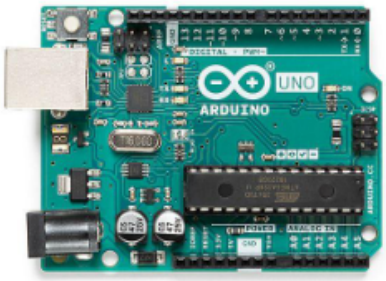


FIGURE 2.5 – Carte Arduino UNO

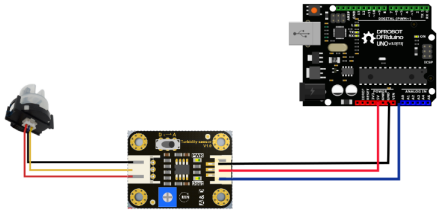


FIGURE 2.6 – Branchement entre la carte arduino et le capteur

2.1.4.2 Langage de programmation et technologies utilisées

React Native

React Native est un framework de développement d'applications mobiles basé sur la librairie web React et utilise le langage de programmation Javascript. Sa particularité est qu'il permet de développer et déployer simultanément votre application pour les plateformes iOS et Android avec une base de code unique . Cela signifie que vous pouvez économiser du temps et de l'argent en n'ayant pas à développer deux applications distinctes pour chaque plateforme. React Native est également très populaire auprès des développeurs car il est facile à apprendre et à utiliser. Il dispose également d'une grande communauté de développeurs qui peuvent vous aider si vous rencontrez des problèmes.

Enfin, React Native est compatible avec les API REST que nous allons utiliser.En effet, ils présentent un moyen courant de communiquer avec les serveurs Web et les bases de données.

Parmis les points fort de ce framework :

— Un soutien solide de la communauté.
— Facilité de maintenance.
— Structure de code stable.
— Rendu rapide.
— Performance optimisée.
— Code réutilisable et partageable.
— Rechargement en direct

En résumé, React Native est un excellent choix pour le développement d'applications mobiles grâce à sa facilité d'utilisation, sa flexibilité, sa capacité à partager du code entre les plateformes et son large soutien communautaire.

Les figures 2.7, 2.8 et 2.9 présentent trois interfaces de notre application, soit la première pour enregistrer ou connecter un utilisateur dans l'application, la deuxième pour capturer une image en gardant uniquement la partie centrale (Région d'intérêt) qui sera envoyée à la base de données, et on finit par l'interface d'écran d'accueil pour un administrateur.

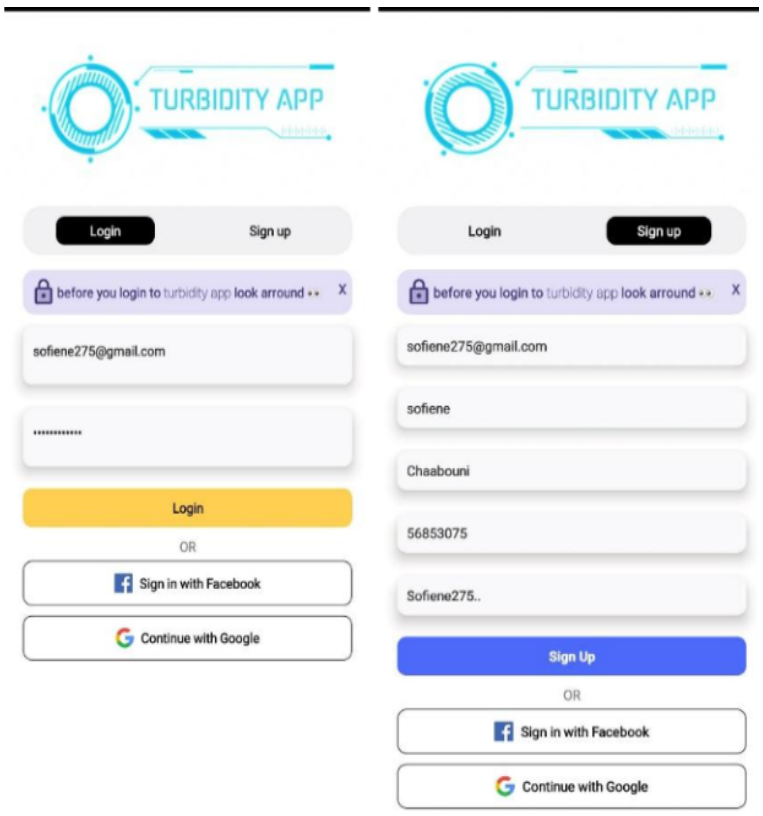


FIGURE 2.7 – Interface d'inscription et de connexion.

FIGURE 2.8 – Interface de capture d'image

FIGURE 2.9 – Interface d'écran d'accueil

Firebase

Firebase est une plateforme de développement d'applications mobiles basée sur le cloud développée par Google, principalement utilisée pour la création d'applications mobiles et Web. Elle fournit divers services tels que l'authentification, la base de données en temps réel, le stockage, l'hébergement et bien plus encore. L'un des avantages clés de Firebase est qu'il élimine le besoin pour les développeurs de construire et de maintenir leur propre infrastructure côté serveur, leur permettant de se concentrer sur la création de l'application elle-même. Les services de base de données en temps réel et de stockage de Firebase facilitent le stockage et la récupération de données, y compris les entiers et les photos, à partir de l'application elle-même. Firebase Storage vous permet de stocker et de récupérer des fichiers tels que des images et des vidéos dans le cloud. Le service de base de données en temps réel de Firebase est une autre option pour stocker des entiers, qui peuvent être facilement accessibles et manipulés à partir du code côté client. La base de données en temps réel est une base de données NoSQL, ce qui signifie qu'elle peut stocker des données au format JSON, la rendant très flexible et facile à utiliser. Dans l'ensemble, Firebase fournit une solution fiable, évolutive et facile à utiliser pour stocker des données et des médias dans le cloud.

La figure 2.10 montre un exemple d'acquisition faite par l'application mobile et chargée dans le Firestore en portant comme nom la valeur de turbidité suivie de la date et du temps .

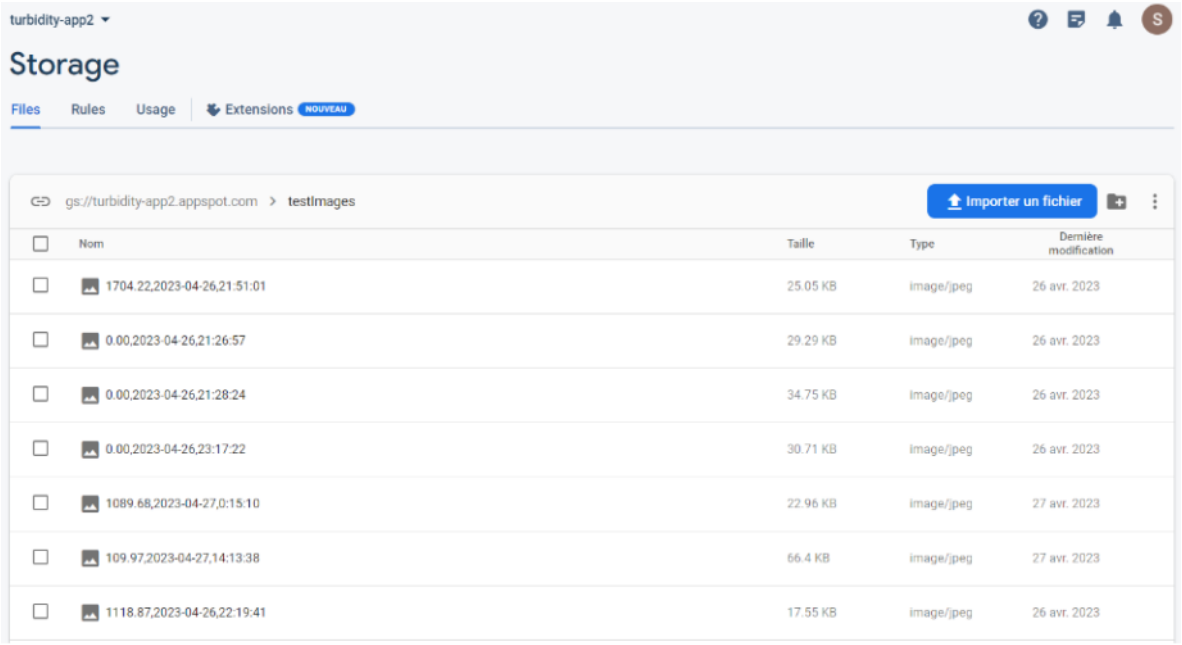


FIGURE 2.10 – Visualisation des images chargées dans le Cloud Firestore.

Flask

Flask est une technologie populaire de développement web qui permet de créer des applications web et des API REST. Flask est un micro-framework basé sur Python, ce qui le rend simple à utiliser et à comprendre, tout en étant très flexible et évolutif. Il utilise des librairies et des outils externes pour des fonctionnalités avancées, comme la base de données, la gestion des sessions, la sécurité, la gestion des formulaires et la mise en cache. Flask offre également une intégration facile avec des outils de développement modernes comme jQuery, React, AngularJS et React Native. Dans le cadre de notre projet, nous avons utilisé Flask pour développer un serveur web qui permet d'écouter les données envoyées par un dispositif électronique connecté à un port série et de les renvoyer en format JSON à une application mobile. Flask offre une grande souplesse pour ce type d'utilisation, en permettant de facilement configurer les routes de l'API et d'ajouter des fonctions pour la manipulation des données.

Pourquoi une API REST ?

Il y a plusieurs raisons pour lesquelles nous avons choisi d'implémenter une solution REST API pour notre projet. Tout d'abord, REST est un style d'architecture qui permet de créer des services web flexibles et évolutifs qui peuvent être utilisés par différents clients. De plus, REST est basé sur HTTP, qui est un protocole standard utilisé par tous les navigateurs web et les serveurs web. Cela signifie que nous n'avons pas besoin d'utiliser des bibliothèques tierces pour communiquer avec le serveur depuis notre application mobile. Enfin, REST est un style d'architecture très populaire et bien documenté qui est utilisé par de nombreuses entreprises et organisations pour créer des services web robustes et évolutifs. En résumé, l'utilisation d'une solution REST API est un choix judicieux pour garantir la flexibilité, la compatibilité et la scalabilité de notre projet.

2.2 Choix des descripteurs et des méthodes de classification

Nous abordons l'étape cruciale de la sélection des descripteurs et des méthodes qui seront utilisés pour l'analyse et la classification des données. Cette étape revêt une importance majeure, car elle détermine la manière dont nous allons extraire les caractéristiques pertinentes de nos données et les utiliser pour effectuer des prédictions et des classifications précises. Dans cette section, nous examinerons en détail les descripteurs que nous avons choisis, tels que les descripteurs de texture, de forme et de couleur, ainsi que les différentes méthodes de classification que nous avons explorées. Notre objectif est de créer un modèle efficace et fiable qui nous permettra de traiter et d'analyser les données de manière optimale.

2.2.1 Choix des descripteurs

2.2.1.1 Les descripteurs de texture

Gray Level Co-occurrence Matrix (GLCM)

La GLCM analyse les relations entre les pixels voisins d'une image et quantifie leur fréquence d'apparition. En tenant compte de la distribution spatiale des intensités des pixels. Puis, sur la base de cette matrice, Haralick propose plusieurs valeurs qui peuvent être extraites de la GLCM pour quantifier la texture selon une direction : la direction horizontale(0 rad). On cite ci-dessous les formules mathématiques utilisées pour extraire quelques indicateurs Haralick :

Correlation : Elle mesure la dépendance linéaire des niveaux de gris de pixels voisins. Elle peut être utile pour détecter des régions linéaires dans une image, où les valeurs des pixels augmentent ou diminuent d'une manière uniforme

$$\text{corrélation} = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(i \cdot j \cdot P(i,j) - \mu_x \cdot \mu_y)}{\sigma_x \cdot \sigma_y}$$

où :
- N est le nombre total de niveaux de gris dans l'image.
- i et j sont les niveaux de gris des deux pixels voisins comparés.
- $P(i,j)$ est la probabilité conjointe d'occurrence des niveaux de gris i et j.
- μ_x et μ_y sont les moyennes des niveaux de gris de l'image pour les pixels x et y respectivement.
- σ_x et σ_y sont les écarts-types des niveaux de gris de l'image pour les pixels x et y respectivement.

Contraste : Il mesure la différence d'intensité entre les pixels adjacents, fournissant ainsi une indication de la netteté des transitions dans l'image.

$$\text{contraste} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(i-\mu)^2 \cdot P(i,j)$$

où :
- N est le nombre total de niveaux de gris dans l'image.
- i et j sont les niveaux de gris des deux pixels voisins comparés.
- μ est la moyenne des niveaux de gris de l'image.
- $P(i,j)$ est la probabilité conjointe d'occurrence des niveaux de gris i et j.

Energie : mesure la répartition et la concentration des niveaux de gris dans l'image, offrant une mesure de la texture globale de l'image.

$$\text{énergie} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P(i,j)^2$$

où :
- N est le nombre total de niveaux de gris dans l'image.
- $P(i,j)$ est la probabilité conjointe d'occurrence des niveaux de gris i et j.

Homogénéité : C'est une mesure de la proximité de la distribution des éléments dans la matrice de co-occurrence à la diagonale de la matrice.

$$\text{homogénéité} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{P(i,j)}{1+|i-j|}$$

où :
- N est le nombre total de niveaux de gris dans l'image.
- i et j sont les niveaux de gris des deux pixels voisins comparés.
- $P(i,j)$ est la probabilité conjointe d'occurrence des niveaux de gris i et j.

Dissimilarité : Elle donne une mesure de la variation des niveaux de gris entre un pixel et son voisinage dans l'image.

$$\text{dissimilarité} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |i-j| \cdot P(i,j)$$

où :
- N est le nombre total de niveaux de gris dans l'image.
- i et j sont les niveaux de gris des deux pixels voisins comparés.
- $P(i,j)$ est la probabilité conjointe d'occurrence des niveaux de gris i et j.

Local Binary Pattern (LPB)

L'algorithme LBP est une méthode simple mais puissante qui caractérise les motifs de texture en comparant les valeurs d'intensité de chaque pixel avec celles de ses voisins.

L'algorithme attribue un code binaire à chaque voisin selon que son intensité est supérieure ou inférieure à celle du pixel central. Ces codes binaires sont ensuite combinés pour former un motif unique qui représente la texture à cet endroit précis. En répétant ce processus pour chaque pixel de l'image, nous construisons un histogramme qui représente la distribution des différents motifs LBP.

On distingue deux paramètres clés du LBP :
— Rayon : à quelle distance on va chercher les points.
— Nombre de points voisins.

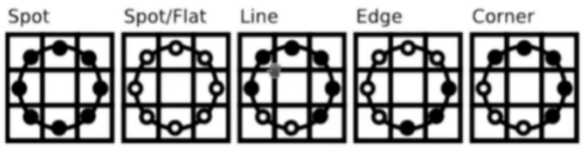


FIGURE 2.11 – Caractérisation locale des régions par LBP

Le pixel central est utilisé comme seuil pour le pixel voisin, et le code LBP d'un pixel central est généré en codant la valeur de seuil calculée en une valeur décimale. L'expression mathématique du LBP est donnée par la formule suivante :

$$LBP(x_c) = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p$$

où :

x_c est le pixel central de l'image,

P est le nombre de pixels voisins dans un voisinage circulaire autour du pixel central,

g_p est la valeur de niveau de gris du pixel voisin p,

g_c est la valeur de niveau de gris du pixel central,

$$s(x) = \begin{cases} 1, & \text{si } x \geq 0 \\ 0, & \text{si } x < 0 \end{cases}$$

2.2.1.2 Les descripteurs de couleur

Histogramme de couleurs

Les histogrammes de couleur sont une caractéristique utile à extraire des images. En calculant les histogrammes de couleur pour chaque image, nous avons pu saisir la distribution des couleurs dans l'image et utiliser cette information pour les différencier avec différents niveaux de turbidité. Par exemple, on peut s'attendre à ce que les images à faible turbidité aient une plus grande proportion de couleurs noires et blanches du motif, alors que les images à forte turbidité ont une plus grande proportion de couleurs brunes.

Moment de couleur

Les moments de couleur constituent une technique précieuse d'extraction des caractéristiques des couleurs pour capturer les informations chromatiques essentielles des images d'eau recadrées. En exploitant les moments de couleur, nous pouvons résumer leur distribution et calculer des mesures statistiques telles que la moyenne, l'écart-type et l'asymétrie des canaux de couleur.

L'utilisation des moments de couleur nous permet d'analyser efficacement les propriétés chromatiques spécifiques à la turbidité de l'eau. En extrayant des mesures statistiques des canaux de couleur, nous pouvons différencier les différents niveaux de turbidité sur la base des motifs de couleur distinctifs présents dans les images. Ces caractéristiques dérivées des moments de couleur donnent un aperçu de la distribution générale des couleurs et des variations d'intensité présentes dans les images d'eau.

Le moment moyen de la couleur représente la valeur moyenne de la couleur, indiquant la tonalité dominante de la couleur dans l'image.

$$\text{moment\_moyen} = \frac{1}{N} \sum_{i=1}^{N} C_i$$

où :
- N est le nombre total de pixels dans l'image.
- C_i est la valeur de la composante de couleur pour le pixel i.

L'écart-type saisit l'étalement ou la variation des couleurs, reflétant la diversité ou l'uniformité des propriétés chromatiques.

$$\text{écart\_type} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(C_i - \text{moment\_moyen})^2}$$

où :
- N est le nombre total de pixels dans l'image.
- C_i est la valeur de la composante de couleur pour le pixel i.
- moment_moyen est le moment moyen de couleur calculé précédemment.

L'asymétrie (skewness) mesure l'asymétrie de la distribution des couleurs, indiquant si certaines couleurs sont surreprésentées ou sous-représentées.

$$\text{asymétrie} = \frac{\frac{1}{N}\sum_{i=1}^{N}(C_i - \text{moment\_moyen})^3}{\left(\frac{1}{N}\sum_{i=1}^{N}(C_i - \text{moment\_moyen})^2\right)^{\frac{3}{2}}}$$

où :
- N est le nombre total de pixels dans l'image.
- C_i est la valeur de la composante de couleur pour le pixel i.
- moment_moyen est le moment moyen de couleur calculé précédemment.

En outre, les moments de couleur offrent une efficacité de calcul et une robustesse aux changements de conditions d'éclairage, qui sont des facteurs cruciaux pour l'analyse en temps réel et la gestion des variations dans les paramètres de capture d'images.

Pour conclure, l'application des moments de couleur dans notre projet nous permet d'extraire des informations chromatiques essentielles des échantillons d'eau. En considérant les mesures statistiques des canaux de couleur, nous obtenons des informations sur la couleur moyenne, la variation des couleurs et l'asymétrie de la distribution des couleurs, contribuant ainsi à la prédiction précise des niveaux de turbidité de l'eau sur la base des caractéristiques des couleurs capturées.

Couleur dominante

La caractéristique de couleur dominante représente la couleur la plus répandue ou la plus proéminente dans l'image. En utilisant des techniques telles que les algorithmes de regroupement ou la quantification des couleurs, nous pouvons identifier les couleurs dominantes présentes dans les échantillons d'eau.

Pour extraire les caractéristiques des couleurs dominantes, nous analysons la distribution des couleurs dans l'image et déterminons la ou les couleurs les plus fréquentes. Ces informations fournissent des indications précieuses sur la tonalité de la couleur dominante, qui peut être révélatrice de l'aspect général et de la composition des échantillons d'eau. La caractéristique de la couleur dominante sert de descripteur représentatif des propriétés chromatiques spécifiques à la turbidité de l'eau. En identifiant les couleurs dominantes, nous pouvons distinguer les différents niveaux de turbidité sur la base des

profils de couleur distinctifs présentés par les échantillons d'eau. En résumé, en identifiant les couleurs les plus répandues, nous pouvons discerner différents niveaux de turbidité sur la base des profils de couleur distinctifs.

2.2.1.3 Les descripteurs de forme

Transformé de Hough

L'utilisation de la transformée de Hough a joué un rôle crucial dans la détection et l'analyse des formes géométriques dans les images d'eau. Nous introduisons un motif composé de cinq rectangles de tailles différentes placés stratégiquement au cours du processus de capture des images d'eau. En appliquant la transformée de Hough, nous avons pu extraire les paramètres des lignes qui correspondent aux bords de ces rectangles, ce qui nous a permis de détecter et de caractériser avec précision leur présence dans les échantillons d'eau. En analysant les équations des lignes obtenues, nous pouvons déterminer avec précision la position, l'orientation et les dimensions de chaque rectangle, ce qui fournit des informations précieuses sur variations de la turbidité de l'eau.

L'incorporation de la transformée de Hough, combinée au motif des rectangles, a amélioré notre capacité à analyser les caractéristiques structurelles des échantillons d'eau. En détectant avec précision les lignes correspondant aux bords des rectangles, nous obtenons des informations précieuses sur la présence et les dimensions des éléments structurels dans l'eau.

Sommairement, l'inclusion de la transformée de Hough dans notre projet nous permet de détecter et d'analyser efficacement les formes géométriques représentées par le motif des rectangles dans les images d'eau. En extrayant les paramètres des lignes et en caractérisant les rectangles, nous obtenons des informations essentielles sur les propriétés structurelles des échantillons d'eau, ce qui contribue à une compréhension globale de la turbidité de l'eau.

Moments de calcul de surfaces

Nous avons utilisé les caractéristiques basées sur les moments comme une technique précieuse pour détecter et mesurer la zone occupée par le motif dans les images d'eau.

Ces moments fournissent une approche robuste pour analyser la distribution spatiale et les caractéristiques de forme du motif dans les images et permettent de caractériser les régions en termes de leur position, de leur étendue, de leur orientation et de leur compacité. Contrairement aux moments de couleur qui se concentrent sur la distribution des valeurs de couleur dans une image ou une région, les moments de calcul de surface fournissent des informations spécifiques sur la taille et la forme des régions.

En appliquant les caractéristiques basées sur les moments, nous avons pu extraire une mesure quantitative qui représente les propriétés spatiales du motif qui est la surface. Cette caractéristique nous a permis de localiser précisément et d'estimer la taille du motif dans les échantillons d'eau.

Cependant, il est important de noter que dans des conditions de turbidité élevée, le motif peut ne plus être visible en raison de la visibilité réduite ou de l'obscurcissement causé par l'augmentation des particules dans l'eau. Par conséquent, les caractéristiques basées sur le moment peuvent ne pas être en mesure de détecter la zone du motif avec précision dans de tels cas.

Brièvement, l'utilisation de caractéristiques basées sur le moment dans notre projet nous a permis d'analyser la zone occupée par le motif dans les échantillons d'eau. Cette approche fournit des informations précieuses sur les caractéristiques spatiales du motif, contribuant à notre compréhension des variations de turbidité de l'eau dans notre projet.

Histogram of Oriented Gradients(HOG)

Nous avons utilisé les caractéristiques de l'histogramme des gradients orientés (HOG) comme une technique précieuse pour l'extraction de caractéristiques basées sur la forme dans les images d'eau. Les caractéristiques HOG capturent les variations locales de gradient d'intensité, qui sont cruciales pour la détection des contours et des structures dans les images.

En appliquant les caractéristiques HOG, nous avons pu extraire les histogrammes des orientations des gradients locaux des régions d'intérêt dans les images d'eau. Ces histogrammes capturent des informations sur la distribution des directions de gradient et fournissent une représentation compacte des caractéristiques de forme.

Dans notre projet, les caractéristiques HOG jouent un rôle important dans la détection et l'analyse des structures géométriques présentes dans les images d'eau. Dans notre cas, les contours du motif sont identifiés à l'aide des orientations de gradient obtenues à partir des caractéristiques HOG.

En outre, les caractéristiques HOG sont résistantes aux variations d'éclairage et aux changements de couleur, ce qui les rend adaptées à l'analyse d'images d'eau où les conditions d'éclairage et de turbidité peuvent varier.

En résumé, l'utilisation de ces caractéristiques dans notre projet nous permet d'extraire des informations sur la forme et les contours des objets présents dans les images d'eau.

2.2.2 Choix des méthodes de classification

Nous avons sélectionné trois méthodes de classification renommées pour estimer la turbidité de l'eau à partir des images capturées : les machines à vecteurs de support (SVM), les forêts aléatoires (Random Forest) et la méthode des k plus proches voisins (KNN).

2.2.2.1 Support Vector Machine

Les SVM sont des algorithmes puissants utilisés pour résoudre des problèmes de classification. Ils cherchent à trouver un hyperplan optimal pour séparer les différentes classes de données dans un espace multidimensionnel. Grâce à leur capacité à gérer des données non linéairement séparables, les SVM sont adaptés pour notre tâche de classification de la turbidité de l'eau, où les frontières de décision peuvent être complexes.

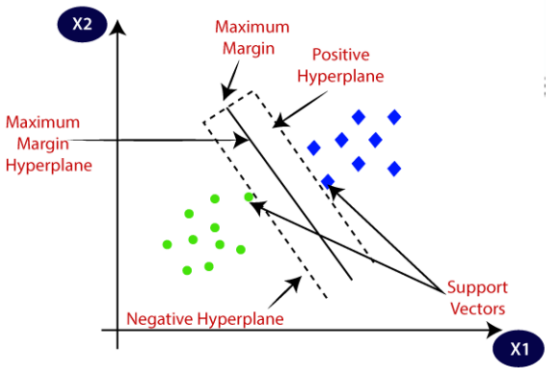


FIGURE 2.12 – SVM : Séparation par un hyperplan entre deux classes selon SVM [4]

2.2.2.2 Random Forest

Les forêts aléatoires sont des méthodes d'apprentissage ensembliste qui combinent les prédictions de plusieurs arbres de décision pour obtenir une prédiction finale. Elles sont réputées pour leur capacité à traiter des ensembles de données de grande taille avec de nombreuses caractéristiques. Les forêts aléatoires sont flexibles, résistantes au surapprentissage et peuvent fournir des estimations robustes de la turbidité de l'eau en exploitant les informations contenues dans les images.

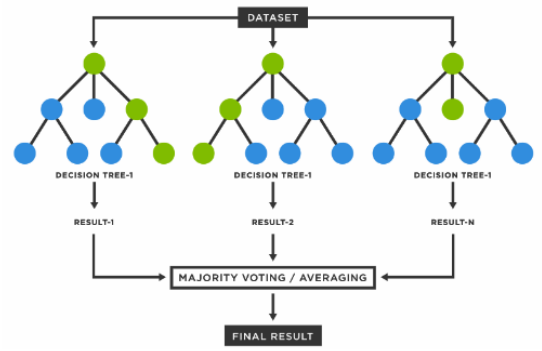


FIGURE 2.13 – Random Forest [5]

2.2.2.3 KNN

La méthode des k plus proches voisins (KNN) est une approche non paramétrique qui se base sur la proximité des échantillons dans l'espace des caractéristiques. En utilisant les k échantillons les plus proches, elle attribue une classe à un nouvel échantillon en fonction de la majorité des classes de ses voisins. Le KNN est simple à mettre en œuvre et peut être efficace lorsque les classes de turbidité présentent des regroupements visuels bien définis dans les images.

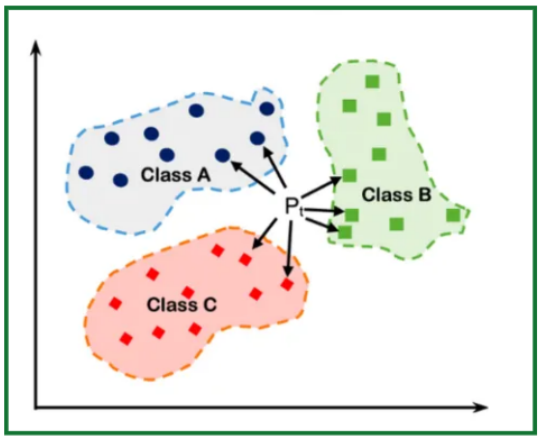


FIGURE 2.14 – KNN [6]

En choisissant ces trois méthodes de classification, nous cherchons à exploiter leurs forces respectives pour obtenir des estimations précises de la turbidité de l'eau. Nous évaluerons les performances de chaque méthode sur notre dataset afin de déterminer laquelle offre les meilleurs résultats en termes de précision et de généralisation. Il est important de souligner que le choix de ces méthodes repose sur leur pertinence et leur efficacité éprouvée dans des problèmes de classification similaires. Cependant, nous restons ouverts à l'exploration d'autres approches de classification si des méthodes prometteuses émergent au cours de notre étude. En conclusion, ces méthodes nous permettront de tirer parti des informations visuelles et des caractéristiques extraites des images pour classer avec précision les échantillons d'eau dans différentes catégories de turbidité.

2.3 Les métriques d'évaluation

Une fois que nous avons entraîné notre modèle à prédire la turbidité de l'eau sur la base des données collectées, il est crucial d'évaluer ses performances à l'aide d'une série de mesures. En évaluant la précision et le rappel du modèle, en analysant la matrice de confusion et en calculant le score F1, nous obtenons des informations complètes sur son efficacité.

2.3.1 Matrice de confusion

Pour mieux comprendre les performances de notre modèle, nous avons effectué une analyse à l'aide d'une matrice de confusion. La matrice de confusion fournit une ventilation complète des prédictions du modèle par rapport aux valeurs réelles de la vérité terrain. Elle nous permet de déterminer le nombre de **faux négatifs** (FN), de **faux positifs** (FP), de **vrais négatifs** (VN) et de **vrais positifs** (VP).

- **Les faux négatifs** se produisent lorsque le modèle prédit à tort des échantillons négatifs comme étant positifs. Il s'agirait alors de classer à tort des niveaux de turbidité élevés comme étant des niveaux de turbidité faibles.
- **Les faux positifs** surviennent lorsque le modèle prédit à tort des échantillons positifs alors qu'ils sont en réalité négatifs. Dans notre projet, il s'agirait de classer à tort les faibles niveaux de turbidité comme étant des turbidités élevées.

— **Les vrais négatifs** représentent les cas où le modèle identifie correctement les échantillons négatifs comme étant négatifs. Dans notre cas, cela correspond à une classification correcte des faibles niveaux de turbidité.
— **Les vrais positifs** correspondent aux cas où le modèle prédit correctement que les échantillons positifs sont positifs. Dans notre contexte, cela signifie qu'il identifie avec précision des niveaux élevés de turbidité dans les échantillons d'eau.

L'analyse de la matrice de confusion nous permet d'identifier des modèles ou des classes spécifiques pour lesquels le modèle peut rencontrer des difficultés.

2.3.2 Exactitude

La mesure de l'exactitude permet d'évaluer l'exactitude globale des prédictions du modèle. Elle représente le rapport entre le nombre d'échantillons correctement classés et le nombre total d'échantillons. Une précision élevée indique que le modèle fait des prédictions précises pour les niveaux de turbidité de l'eau sur la base des données d'entrée.

$$\textbf{Exactitude} = \frac{\text{VP+VF}}{\text{VP+VN+FN+FP}}$$

2.3.3 La précision

mesure la proportion d'éléments correctement classés pour une classe donnée :

$$\textbf{Précision} = \frac{VP}{VP + FP}$$

La précision permet d'évaluer le coût des faux positifs, c'est-à-dire des éléments détectés par erreur. Si l'objectif est de limiter les faux positifs, on cherchera à maximiser cet indicateur.

2.3.4 Le rappel

mesure la proportion d'éléments correctement classés par rapport au nombre total d'éléments de la classe à prédire :

$$\textbf{Rappel} = \frac{VP}{VP + FN}$$

2.3.5 Le score F1

est une mesure de compromis entre la précision et le rappel :

$$\textbf{Score F1} = \frac{2 \times (\text{Précision} \times \text{Rappel})}{\text{Précision} + \text{Rappel}}$$

Ces indicateurs sont utiles pour évaluer les performances d'un modèle de classification et prendre des décisions éclairées sur les mesures à prendre pour optimiser les résultats.

2.4 Conclusion

Pour conclure, l'ensemble de notre processus de développement a été guidé par des choix technologiques réfléchis et des décisions stratégiques. Nous avons veillé à sélectionner les outils, les langages de programmation, les frameworks et les modèles d'apprentissage appropriés pour atteindre nos objectifs. Ces choix ont été faits en fonction de critères tels que la compatibilité avec nos besoins, la robustesse, la facilité d'utilisation et la scalabilité du système.

Chapitre 3

Expérimentation et résultats

Dans ce chapitre, nous présentons la méthodologie employée pour mener les expériences, le processus de collecte des données et les techniques d'analyse utilisées pour interpréter les résultats.

L'objectif principal de ce chapitre est de valider la fonctionnalité et la précision de notre système en évaluant sa capacité à prédire les niveaux de turbidité à partir des images capturées. Grâce à une série d'expériences soigneusement conçues et à une analyse rigoureuse des données, nous visons à déterminer la fiabilité et la précision de notre modèle prédictif.

3.1 Protocole d'acquisition mis en place

Comme montré dans la figure 3.1, le protocole d'acquisition utilisé dans cette étude est soigneusement élaboré pour garantir des résultats précis et fiables. Tout d'abord, les acquisitions sont réalisées dans une chambre avec un niveau de luminosité constant, un verre transparent rempli d'eau qui va servir pour préparer le mélange et avec un motif placé en arrière-plan. De plus, on va utiliser un smartphone équipé d'une caméra de 64 mégapixels est fixé sur une position pour capturer les images à l'aide de l'application mobile qui dès qu'on prend une photo, le turbidimètre renvoie la valeur NTU a un serveur local qui se charge de renvoyer cette valeur a notre application mobile .Cette dernière se charge de sauvegarder l'image avec sa valeur de turbidité . Concernant le motif, il est constitué de quatre traits noirs horizontaux de largeurs de plus en plus petites. Imprimé sur un papier et collé à la face arrière du verre au moment de l'acquisition, le motif est supposé donner plus de dimensions (features) , car au fur et à mesure que l'eau augmente en turbidité, les traits deviennent moins nets, augmentent en contraste et leurs bords deviennent moins acérés. Le processus d'acquisition commence par l'ajout progressif de petites quantités de sédiments très argileux dans l'eau du verre. Ensuite, le liquide est soigneusement mélangé et on attend environ 30 secondes pour que l'échantillon se stabilise avant de prendre la photo.

Ce protocole d'acquisition nous a permis d'avoir différentes classes comme illustré dans les figures 3.2, 3.3 et 3.4.

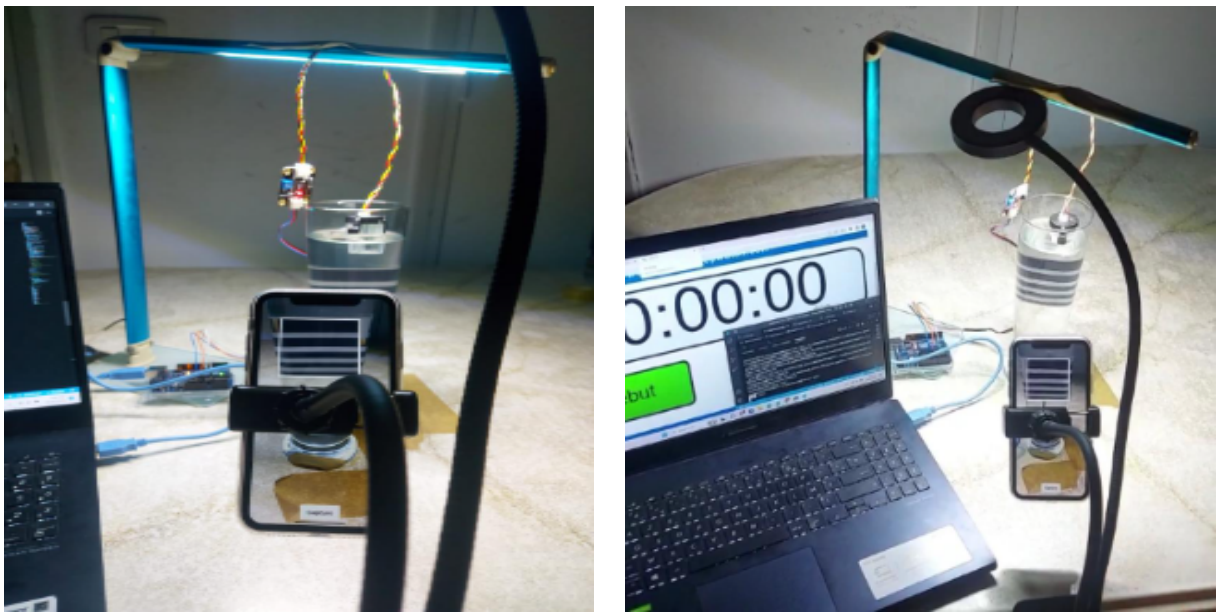
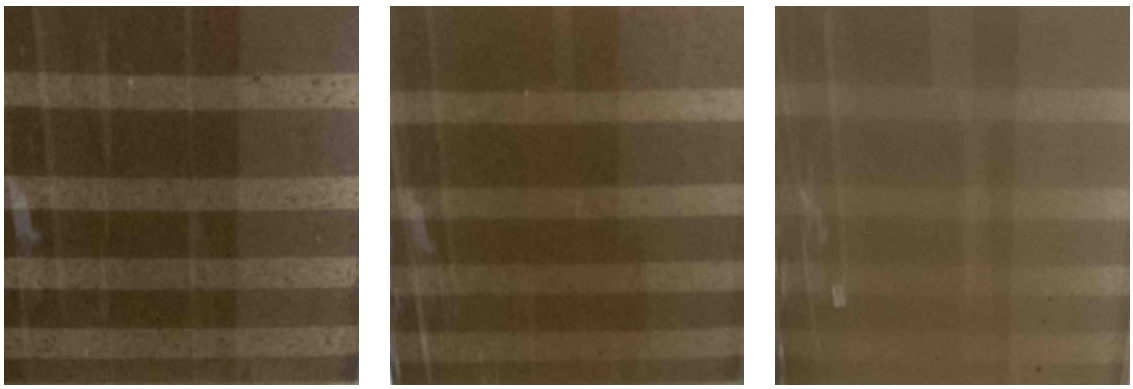


FIGURE 3.1 – Mise en oeuvre du protocole d'acquisition

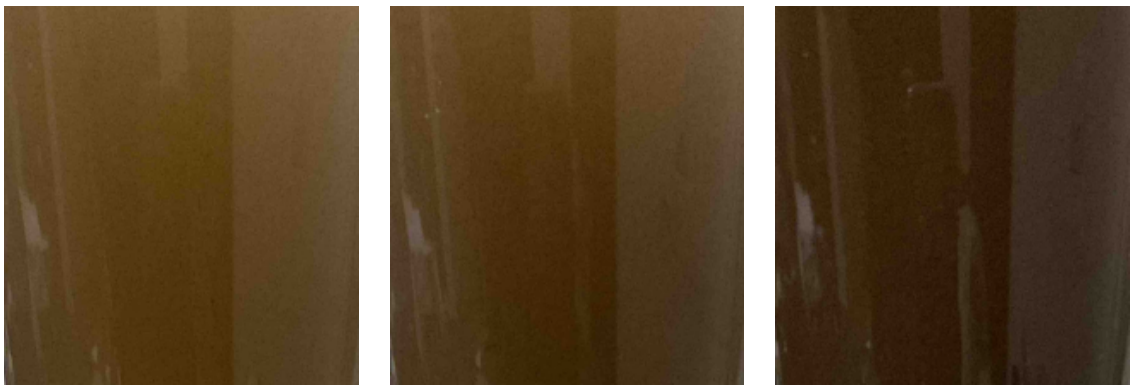
(a) 0 ntu (b) 20 ntu (c) 124 ntu

FIGURE 3.2 – Première classe



(a) 156 ntu (b) 354 ntu (c) 639 ntu

FIGURE 3.3 – Deuxième classe



(a) 1311 ntu (b) 2282 ntu (c) 2300 ntu

FIGURE 3.4 – Troisième classe

3.2 Dataset obtenu

Suite à l'analyse détaillée des différentes valeurs présentes dans notre ensemble de données, nous avons procédé à une binarisation en utilisant 15 bins. Cela nous a permis de générer l'histogramme présent dans la figure 3.5, illustrant la répartition du nombre de photos par intervalle de valeurs. Cette visualisation offre une perspective précieuse sur la distribution des données dans notre jeu de données.

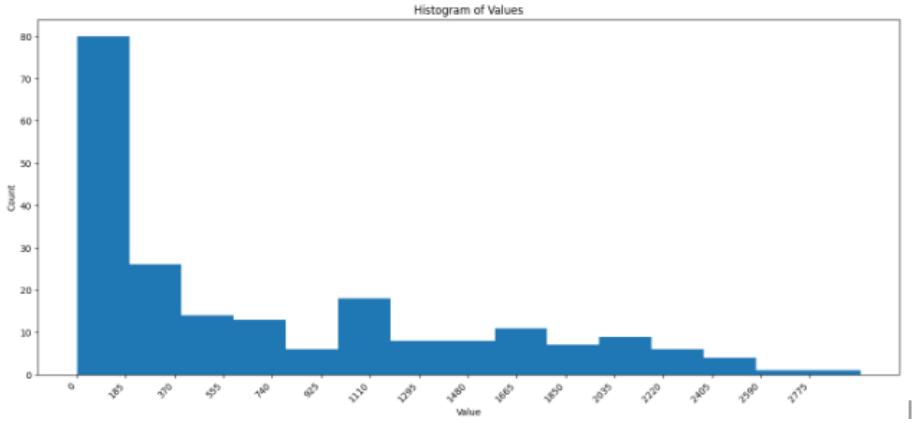


FIGURE 3.5 – Histogramme montrant le nombre d'images par bin.

Après la répartition qu'on a fait pour obtenir 3 classes et donc 3 intervalles on a obtenu l'histogramme de la figure 3.6 montrant le nombre de photos par classe :

Cette figure nous montre la répartition du dataset qui est comme suit :

— Classe 1 : 73 valeurs $<$ 150 NTU
— Classe 2 : 51 valeurs $>$ 150 et $<$ 700 NTU
— Classe 3 : 88 valeurs $>$ 700 NTU

Si on compare ces chiffres, il est clair que la distribution des échantillons entre les différentes classes n'est pas parfaitement équilibrée. La classe 3 a le plus grand nombre d'échantillons, et la classe 2 a le plus petit nombre. Cependant, la différence n'est pas extrêmement grande.

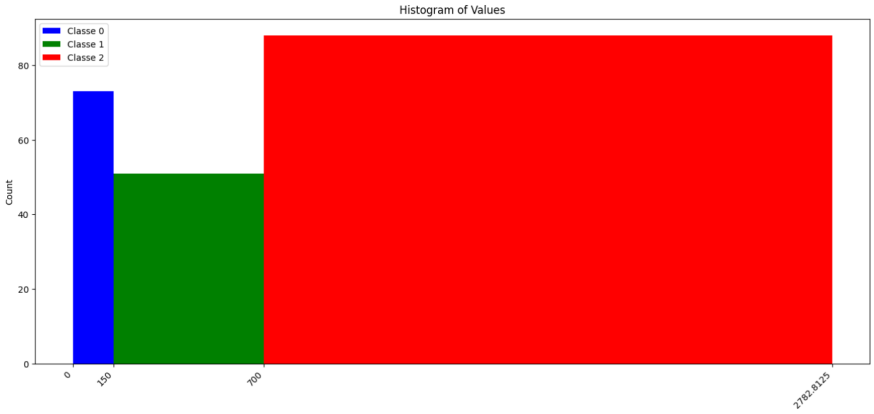


FIGURE 3.6 – Histogramme montrant le nombre d'images par classe.

3.2.1 Normalisation

Afin de garantir des comparaisons justes et précises entre les caractéristiques, nous avons effectué une normalisation spécifique à chaque caractéristique. Dans cette approche, nous avons normalisé individuellement les valeurs de chaque descripteur : forme, couleur et texture.
En traitant chaque caractéristique séparément, le processus de normalisation s'appuie uniquement sur les valeurs observées dans cette caractéristique spécifique.

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

X : La valeur initiale d'une donnée spécifique de la feature.
X_{min} : La valeur minimale observée dans cette feature.
X_{max} : La valeur maximale observée dans cette feature.
X_{norm} : La valeur normalisée de la donnée spécifique de la feature.

Cette stratégie nous a permis d'éliminer tout biais ou écart potentiel résultant de variations dans l'échelle ou l'étendue des caractéristiques.

En normalisant les valeurs de chaque caractéristique, nous avons assuré qu'aucune caractéristique ne domine le processus d'apprentissage et que le modèle peut extraire efficacement les informations pertinentes des caractéristiques de forme, de couleur et de texture.

L'étape de normalisation joue un rôle essentiel dans notre projet, car elle nous a aidé à créer des conditions égales pour les différentes caractéristiques et a favorisé des prédictions précises et fiables de la turbidité de l'eau sur la base des seules images capturées. En incorporant cette technique de normalisation dans notre flux de travail de prétraitement des données, nous avons amélioré la qualité et la comparabilité des descripteurs, ce qui contribue à la robustesse et à la performance de notre modèle prédictif.

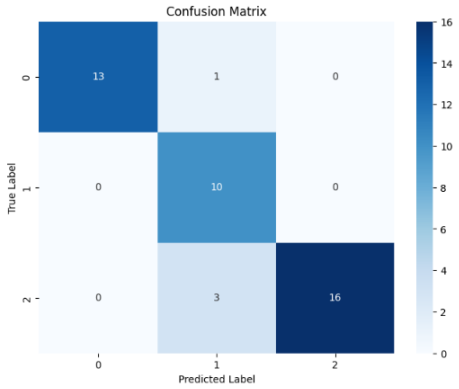
3.3 Résultats de la classification

Lors de l'utilisation des modèles, nous avons réservé 20% des images du dataset pour la validation, ce qui correspond à 43 images. Les 80% restants, soit 169 images,

ont été utilisés pour l'entraînement des modèles. Cette séparation entre les données d'entraînement et de validation nous permet d'évaluer les performances des modèles sur des données non vues auparavant et d'obtenir une estimation réaliste de leur précision

3.3.1 Random Forest

Dans un premier temps, nous avons mis en œuvre un modèle simple basé sur l'algorithme de la forêt aléatoire (Random Forest). Les valeurs manquantes dans le jeu de données ont été gérées par l'imputation de la moyenne, ce qui signifie que nous avons remplacé les valeurs manquantes par la moyenne des valeurs existantes dans chaque colonne. Avec cette approche, le modèle a produit une exactitude(accuracy) de 0.9 comme il est montré dans la figure 3.7.



Matrice de confusion

CLASSE	PRÉCISION	RAPPEL	F1-SCORE	SUPPORT
Turbidité Faible	1.0	0.928	0.963	14
Turbidité Moyenne	0.714	1.0	0.833	10
Turbidité élevée	1.0	0.842	0.914	19
Macro AVG			0.903	43
Weighted AVG			0.911	43

Rapport

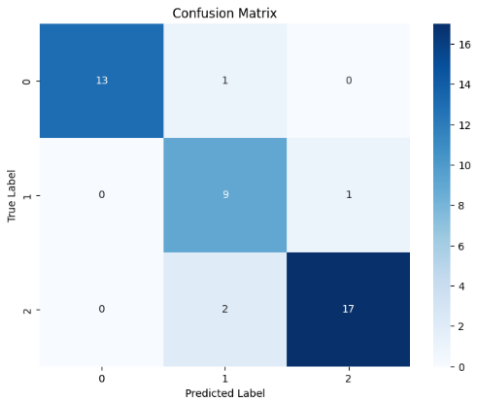
FIGURE 3.7 – Matrice de confusion et Rapport du modèle random forest utilisant l'imputation par moyenne.

L'exactitude globale du modèle utilisant l'imputation des valeurs moyennes est de 90.7%. Cette mesure représente la proportion des prédictions correctes parmi le total des prédictions.

Les colonnes "Précision", "Rappel" et "F1-Score" représentent respectivement la précision, le rappel et le score F1 pour chaque classe, ainsi que leur moyenne macro (moyenne simple des valeurs pour chaque classe) et moyenne pondérée (moyenne des valeurs pour chaque classe, pondérée par le nombre de vrais exemples pour chaque classe).

La colonne "Support" représente le nombre de vrais exemples pour chaque classe dans l'ensemble de données. Cela peut être utile pour comprendre si les mesures de performance sont influencées par le déséquilibre des classes.

Ce résultat s'est également manifesté lorsque nous avons utilisé uniquement les caractéristiques qui n'avaient aucune valeur manquante comme il est indiqué dans la figure 3.8.



Matrice de confusion

CLASSE	PRÉCISION	RAPPEL	F1-SCORE	SUPPORT
Turbidité Faible	1	0.93	0.96	14
Turbidité Moyenne	0.75	0.90	0.82	10
Turbidité élevée	0.94	0.89	0.92	19
Macro AVG			0.90	43
Weighted AVG			0.91	43

Rapport

FIGURE 3.8 – Matrice de confusion et Rapport du modèle random forest utilisant des features qui n'ont pas de valeurs manquantes.

L'exactitude globale du modèle est de 90.7%. Cette mesure représente la proportion des prédictions correctes parmi le total des prédictions.

Séléction des caractéristiques

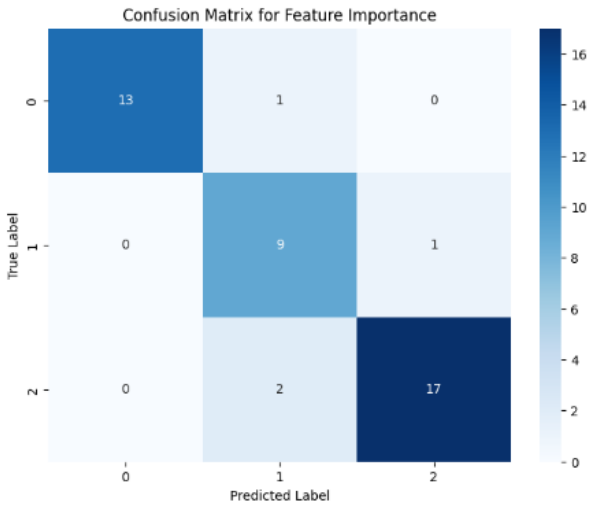
Nous avons mis en œuvre deux méthodes différentes pour la sélection de caractéristiques avec l'objectif d'améliorer l'exactitude de notre modèle de classification.

Importance des caractéristiques : C'est une approche intégrée dans les modèles basés sur les arbres, tels que le RandomForest. Cette méthode assigne un score à chaque caractéristique en se basant sur la mesure dans laquelle elle aide à améliorer la précision de la prédiction. Nous avons utilisé cette méthode pour sélectionner les 12, 20 et 30 caractéristiques les plus importantes.

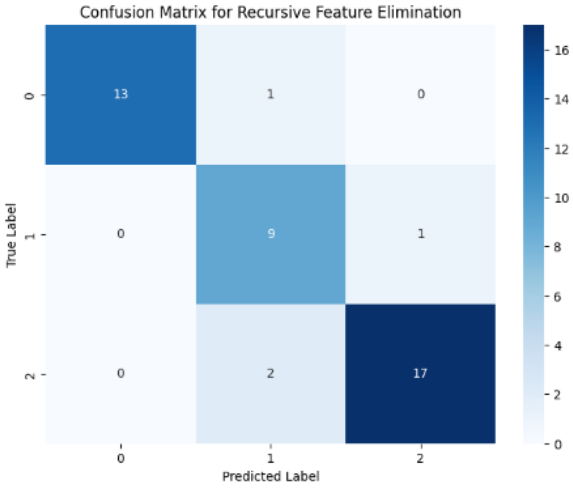
Élimination récursive des caractéristiques (RFE) : RFE est une méthode de sélection de caractéristiques qui fonctionne en éliminant successivement les caractéristiques les moins importantes. Elle utilise le modèle pour classer l'importance des caractéristiques, puis élimine les caractéristiques les moins importantes et réitère le processus jusqu'à ce qu'un nombre spécifié de caractéristiques soit atteint. Dans notre cas, nous avons utilisé RFE pour réduire le nombre de caractéristiques aux 12, 20 et 30 meilleures .

Pour la sélection de 20 caractéristiques principales avec l'importance des caractéristiques, nous avons atteint une exactitude (accuracy) de 0.91 et un score F1 de 0.91 à partir de la matrice de confusion de la figure 3.9 . Ces résultats se sont révélés supérieurs à ceux obtenus pour les cas de 12 et 30 caractéristiques.

En sélectionnant 20 meilleures caractéristiques principales avec RFE, nous avons obtenu une exactitude (accuracy) de 0.91 et une moyenne pondérée pour le score F1 de 0.91 à partir de la matrice de confusion de la figure 3.9. Cela suggère que la sélection de 20 ou 12 caractéristiques principales offre une meilleure performance en termes d'exactitude (accuracy) et de score F1 par rapport à la sélection de 30 caractéristiques. Les 12 meilleures caractéristiques que nous avons identifiées sont principalement liées aux surfaces (areas) de l'image et aux caractéristiques de couleur. Ces traits semblent jouer un rôle important dans notre modèle.



Matrice de confusion
avec importance des caractériqtiques



Matrice de confusion
avec RFE.

FIGURE 3.9 – Comparaison des matrices de confusion du modèle Random forest avec séléction des caractéristiques.

NOMBRE DES MEILLEURES CARACTÉRISTIQUES	12	20	30
CLASEE 0			
Précision	1.00	1	1
F1-score	0.96	0.96	0.96
CLASEE 1			
Précision	0.73	0.75	0.73
F1-score	0.76	0.82	0.76
CLASEE 2			
Précision	0.89	0.94	0.89
F1-score	0.89	0.92	0.89

Performance avec importance des caractériqtiques

NOMBRE DES MEILLEURES CARACTÉRISTIQUES	12	20	30
CLASEE 0			
Précision	1.00	1	1
F1-score	0.96	0.96	0.96
CLASEE 1			
Précision	0.75	0.75	0.73
F1-score	0.82	0.82	0.76
CLASEE 2			
Précision	0.94	0.94	0.89
F1-score	0.92	0.92	0.89

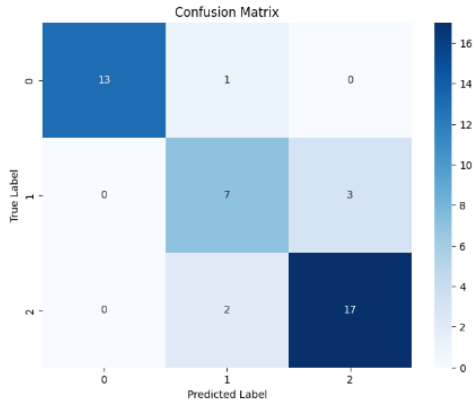
Performance avec RFE.

FIGURE 3.10 – Comparaison des performances des modèles Random Forest avec séléction des caractéristiques.

Ces deux méthodes ont montré des résultats similaires, comme le démontre les figures 3.9 et 3.10.
Nous avons exploré deux méthodes de sélection de caractéristiques, à savoir RFE (Recursive Feature Elimination) et l'importance des caractéristiques, afin d'optimiser notre modèle. Malgré nos efforts, nous n'avons pas observé d'améliorations significatives dans les résultats obtenus. Il est possible que la structure de nos données limite l'efficacité de ces approches de sélection de caractéristiques. Cela pourrait également indiquer que notre modèle est déjà à un point d'optimisation et que des améliorations supplémentaires nécessiteraient des techniques de modélisation plus avancées ou un réajustement plus fin des paramètres.

3.3.2 KNN

La sélection d'un nombre approprié de voisins (k) est crucial pour obtenir de bonnes performances de l'algorithme k-NN. Pour cela, nous avons mis en œuvre une recherche en grille (GridSearchCV), qui est une méthode d'optimisation des hyperparamètres qui effectue systématiquement une recherche exhaustive à travers un ensemble prédéfini de valeurs des hyperparamètres. Notre recherche en grille a été configurée pour tester les valeurs de k allant de 1 à 20. Après avoir entraîné le modèle k-NN avec chaque valeur de k sur notre jeu d'entraînement, nous avons utilisé la validation croisée pour évaluer la performance du modèle. La validation croisée est une technique qui fournit une évaluation robuste des performances d'un modèle en le formant et en le testant sur différentes partitions de l'ensemble de données. La recherche en grille a identifié le meilleur k qui est égal à 17 pour une répartition sur 5 dans la cross validation. En utilisant cette valeur optimale de k, nous avons ensuite entraîné notre modèle k-NN final et obtenu une accuracy 0.86 sur le jeu de test, ce qui indique que notre modèle était capable de prédire correctement 86% des instances dans les données de test. Ce processus nous a permis d'optimiser de manière efficace notre modèle k-NN et d'obtenir un modèle avec une performance de prédiction élevée comme montré dans la figure 3.11.



Matrice de confusion

CLASSE	PRÉCISION	RAPPEL	F1-SCORE	SUPPORT
Turbidité Faible	1.0	0.93	0.96	14
Turbidité Moyenne	0.70	0.70	0.70	10
Turbidité élevée	0.85	0.89	0.87	19
		Macro AVG	0.84	43
		Weighted AVG	0.86	43
			Accuracy	0.86

Rapport

FIGURE 3.11 – Matrice de confusion et Rapport du modèle KNN.

Dans le cas d'une division en 10 sous-ensembles (folds), l'exactitude (accuracy) est de 0.81 pour une valeur optimale de k égale à 10.

Dans le cas d'une division en 50 sous-ensembles (folds), l'exactitude (accuracy) est de 0.84 pour une valeur optimale de k égale à 11.

Dans le cas d'une division en 30 sous-ensembles (folds), l'exactitude (accuracy) est de 0.86 pour une valeur optimale de k égale à 16.

NOMBRE DE SOUS-ENSEMBLES	5	10	30	50
Accuracy	0.86	0.81	0.86	0.84

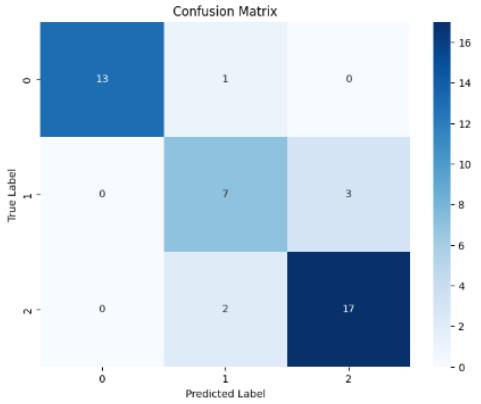
FIGURE 3.12 – Performances des différents modèles avec des sous-ensembles d'entraînement différents.

Nous avons opté pour le modèle qui affiche la meilleure exactitude (le cas de 5 sous-ensembles) parmi les options proposées dans la figure 3.12. Même s'il présente des

résultats similaires à un autre modèle utilisant un nombre de sous-ensembles (folds) plus élevé, ce choix est avantageux en termes de performances. En effet, l'utilisation d'un nombre de folds inférieur se traduit par une réduction du temps de calcul et une optimisation des ressources, tout en maintenant une précision élevée. C'est donc une solution plus efficace et efficiente.

3.3.3 Support Vector Machine

Dans un premier temps nous avons développé un modèle simple basé sur l'algorithme SVM. Nous avons utilisé la bibliothèque SVC avec ses hyperparamètres par défaut, et les résultats obtenus sont illustrés dans la figure 3.13. Ce modèle initial a produit une exactitude (accuracy) de 0.86 et un score F1 global de 0.86.



Matrice de confusion

CLASSE	PRÉCISION	RAPPEL	F1-SCORE	SUPPORT
Turbidité Faible	1.0	0.93	0.96	14
Turbidité Moyenne	0.70	0.70	0.70	10
Turbidité élevée	0.85	0.895	0.88	19
		Macro AVG	0.85	43
		Weighted AVG	0.86	43
			Accuracy	0.86

Rapport

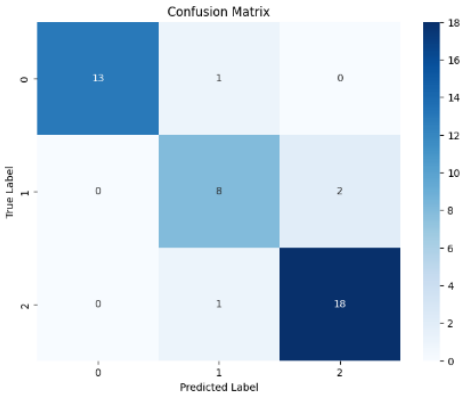
FIGURE 3.13 – Matrice de confusion et Rapport du modèle SVM.

Ce modèle montre des résultats très similaires à ceux du modèle KNN. Ces résultats peuvent être expliqués par les données qui sont linéairement séparables, en plus de la petite taille du dataset. Ajoutons à cela qu'on a éliminé tout bruit et données aberrantes, ce qui peut aussi expliquer pourquoi on obtient les mêmes résultats.

Dans le cadre de l'optimisation de notre modèle SVM (Machine à Vecteurs de Support), nous avons utilisé l'outil GridSearchCV de la bibliothèque Sklearn. Cet outil a pour fonction de rechercher de manière exhaustive la meilleure combinaison des hyperparamètres à partir d'une grille prédéfinie. Dans notre cas, nous avons exploré diverses valeurs pour 'C'(paramètre de régularisation). L'objectif de cette démarche était d'équilibrer le biais et la variance du modèle, ainsi que d'ajuster l'influence de chaque exemple d'entraînement. Après avoir entraîné GridSearchCV sur les données, nous avons pu déterminer la combinaison des hyperparamètres qui a donné les meilleures performances de validation croisée. Nous avons ensuite utilisé cette combinaison optimale pour réaliser des prédictions sur mon ensemble de test et évaluer la précision globale de notre modèle optimisé .

En pratique, le choix de 'C' peut avoir un impact important sur la performance du modèle SVM et nécessite généralement un réglage attentif via la validation croisée, comme nous l'avons fait.Nous avons obtenu un meilleur 'C' égal a 10 parmi les valeurs 0.1 et 100 et 1000 .

Les résultats du modèle amélioré obtenu sont présentes dans la figure 3.14 .



Matrice de confusion

CLASSE	PRÉCISION	RAPPEL	F1-SCORE	SUPPORT
Turbidité Faible	1.0	0.93	0.96	14
Turbidité Moyenne	0.8	0.8	0.8	10
Turbidité élevée	0.9	0.95	0.92	19
		Macro AVG	0.90	43
		Weighted AVG	0.91	43
			Accuracy	0.91

Rapport

FIGURE 3.14 – Matrice de confusion et Rapport du modèle SVM avec une meilleure valeur d'hyperparamètre C.

Après avoir amélioré le modèle SVM à l'aide de la validation croisée, nous observons une nette amélioration des performances. L'exactitude (accuracy) passe de 0.86 à 0.91, indiquant une meilleure prédiction globale. De même, les scores F1 pour toutes les classes se sont améliorés, signifiant un meilleur équilibre entre précision et rappel. En somme, l'optimisation par validation croisée a permis d'améliorer la qualité des prédictions du modèle.
Nous allons donc utiliser ce modèle amélioré puisqu'il présente de meilleurs résultats.

3.3.4 Discussion

Le tableau de la figure 3.15 offre une vue d'ensemble des résultats obtenus avec les différents modèles que nous avons choisis. D'après nos observations, le modèle Random Forest avec imputation des valeurs moyennes affiche les meilleures performances, ceci en prenant en compte l'ensemble des métriques comparées, suivi du modèle Random forest utilisant la méthode RFE, puis du modèle SVM.

MODÈLE	ACCURACY	mAP	WEIGHTED F1-SCORE
Random forest avec l'imputation par moyenne	0.90	0.85	0.91
Random forest avec les caractéristiques non manquantes	0.90	0.81	0.91
Random forest avec sélection de 20meilleures features avec RFE	0.91	-	0.91
Random forest avec sélection de 20meilleures features avec l'importance des caractéristiques	0.91	-	0.91
KNN	0.86	0.77	0.86
SVM	0.91	0.77	0.91

FIGURE 3.15 – Tableau comparatif des différents modèles testés.

Ces résultats pourraient être améliorés, mais quelques facteurs ont contribué à une performance moindre. Premièrement, la taille de base de données est assez petite. Un ensemble de données plus grand pourrait aider à affiner le modèle et à améliorer sa capacité de généralisation.

Deuxièmement, les valeurs de turbidité fournies par le capteur ne sont pas précises. En effet, le capteur utilisé n'est pas un véritable turbidimètre, ce qui a entraîné des imprécisions dans les mesures et ainsi introduit du bruit dans l'ensemble de données.

3.4 Résultats de classification après augmentation des données

3.4.1 Nouvelles acquisitions

Pour rendre notre modèle plus généralisé, nous avons choisi d'élargir notre dataset en ajoutant 109 nouvelles photos, principalement des exemples de classes de faible et moyenne turbidité, comme illustré dans la figure 3.16 qui présente l'histogramme de répartition du nombre de photos selon les intervalles de valeurs. Cette expansion du dataset nous permet d'avoir une meilleure représentation des différentes catégories de turbidité et d'améliorer ainsi la capacité de notre modèle à se généraliser et à prendre des décisions précises sur de nouvelles données.

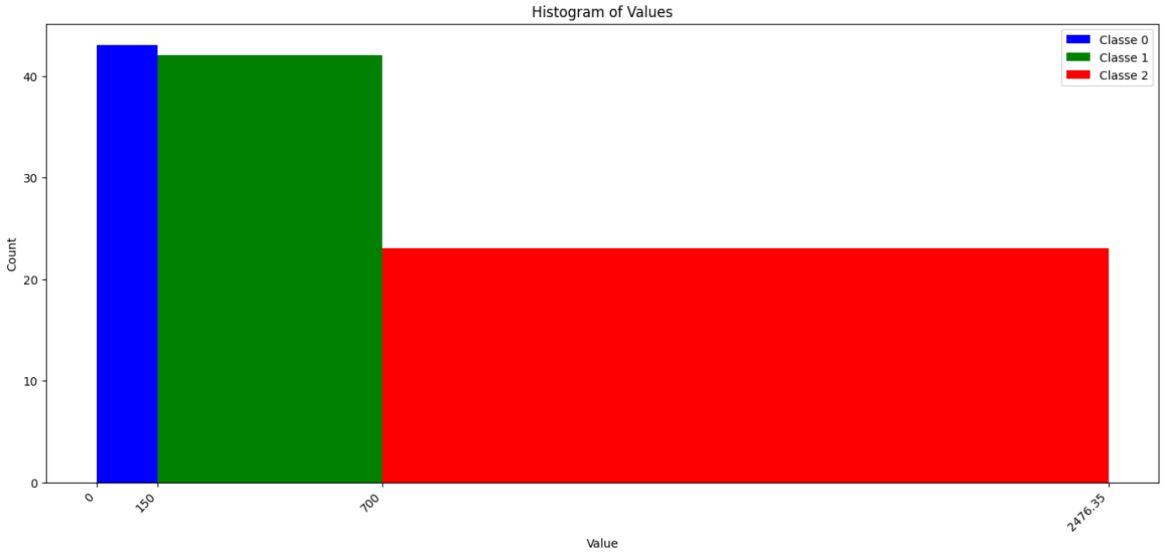


FIGURE 3.16 – Histogramme montrant le nombre d'images par classe.

Après avoir effectué la concaténation du nouveau dataset avec l'ancien dataset et appliqué un mélange aléatoire sur l'ensemble des données, nous avons obtenu la répartition suivante : 44 photos pour la classe 0, 43 photos pour la classe 1 et 23 photos pour la classe 2.

Le Dataset final est représenté dans la figure 3.17.

CLASSE	NOMBRE D'IMAGES
Turbidité Faible	117
Turbidité Moyenne	94
Turbidité élevée	111

FIGURE 3.17 – Tableau montrant le nombre d'images par classe.

Cette augmentation du dataset nous permet d'avoir une meilleure diversité d'échantillons et de renforcer la représentativité de chaque classe et donc un dataset plus ou moins équilibré (où on a toujours un problème de sous-effectif dans la classe 1), ce qui est essentiel pour améliorer les performances et la généralisation de notre modèle.

Lors de l'utilisation des modèles, nous avons réservé 20% des images du dataset pour la validation, ce qui correspond à 64 images. Les 80% restants, soit 256 images, ont été utilisés pour l'entraînement des modèles. Cette séparation entre les données d'entraînement et de validation nous permet d'évaluer les performances des modèles sur des données non vues auparavant et d'obtenir une estimation réaliste de leur précision.

Nous avons évalué les performances de différents modèles pour la prédiction de la turbidité de l'eau. Nous avons utilisé deux approches avec le modèle Random Forest : l'une en remplaçant les valeurs manquantes par la moyenne des caractéristiques , l'autre en utilisant random forest avec sélection de 30 meilleures caractéristiques avec la méthode RFE. Nous avons également utilisé le modèle SVM fourni par la bibliothèque scikit-learn. Les résultats obtenus sont présentés dans les figures 3.18 et 3.19, qui illustrent les performances de chaque modèle évalué.

MODÈLE	ACCURACY	mAP	F1-SCORE MOYEN
Random forest avec l'imputation par moyenne	0.88	0.8	0.86
Random forest avec sélection de 30 meilleures features avec RFE	0.86	0.77	0.85
SVM	0.84	0.76	0.84

FIGURE 3.18 – Tableau comparatif des différents modèles utilisés.

MODÈLE	F1-SCORE POUR LA CLAASE0	F1-SCORE POUR LA CLAASE1	F1-SCORE POUR LA CLAASE2
Random forest avec l'imputation par moyenne	0.96	0.76	0.87
Random forest avec sélection de 30 meilleures features avec RFE	0.96	0.78	0.82
SVM	0.96	0.76	0.79

FIGURE 3.19 – Tableau comparatif de F1-score pour chaque classe.

On peut constater que le modèle Random Forest avec imputation des valeurs moyennes a montré de meilleurs résultats par rapport aux autres modèles. En effet, il a une exactitude (accuracy) de 0.88. Ce modèle est capable de prédire avec précision les photos de la classe 0 et de la classe 2. Cependant, pour la classe 1, le modèle présente un F1-Score de 0.76, ce qui peut être dû à l'imprécision du capteur de turbidité et à l'ajout de photos bruitées dans cette classe.

Les figures 3.20 et 3.21 représentent les matrices de confusion des deux modèles Random Forest : le premier utilisant l'imputation des valeurs moyennes et l'autre utilisant la méthode RFE sur les 30 meilleures caractéristiques. Ces matrices de confusion permettent de visualiser les performances de chaque modèle en termes de prédictions correctes et incorrectes pour chaque classe .

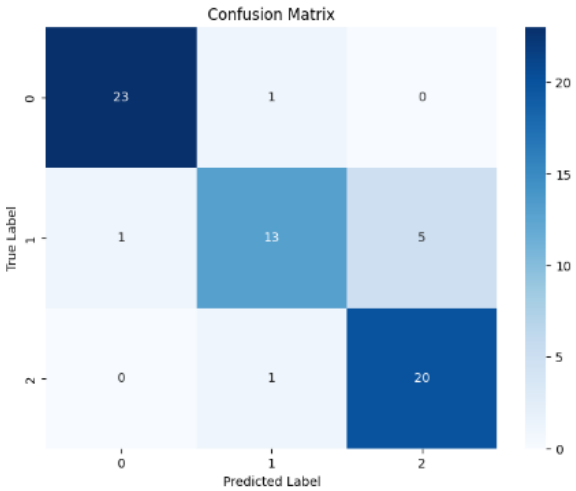


FIGURE 3.20 – Matrice de confusion du modèle Random Forest avec imputation des valeurs moyennes.

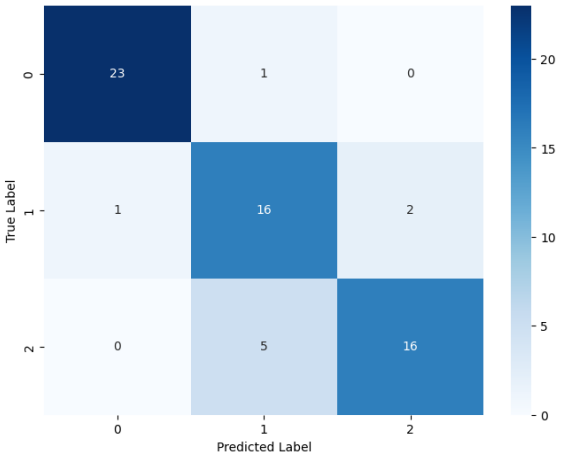


FIGURE 3.21 – Matrice de confusion du modèle Random Forest utilisant les 30 meilleures caractéristiques avec RFE.

3.4.2 Augmentation des données

L'augmentation des données consiste à générer des échantillons de formation supplémentaires en appliquant diverses transformations aux données existantes, telles que des rotations, des retournements et l'ajout de bruit. Cette approche peut contribuer à atténuer le sur-ajustement et à améliorer la capacité du modèle à se généraliser à des données inédites, améliorant ainsi ses capacités prédictives.

Dans le processus de data augmentation de notre dataframe, nous avons utilisé deux méthodes pour varier la texture et le contraste des images. La première méthode consiste à appliquer un filtre de texture à l'aide d'un noyau prédéfini. Dans notre code, nous avons utilisé un filtre de texture qui utilise un noyau spécifique (kernel) pour modifier les caractéristiques de l'image. Cela permet de créer des variations dans la texture de l'image, ce qui peut être utile pour améliorer la robustesse et la généralisation de nos modèles de prédiction.

La deuxième méthode que nous avons utilisée est l'égalisation d'histogramme. Cela implique de convertir l'image en niveaux de gris, puis d'appliquer l'égalisation d'histogramme pour ajuster la distribution des niveaux de gris dans l'image. Cette technique permet d'améliorer le contraste de l'image et de rendre les détails plus visibles.

En utilisant ces deux méthodes, nous avons pu générer de nouvelles images à partir des images originales du dataframe. Les images augmentées ont été ajoutées au dataframe, ce qui a permis d'augmenter la taille du dataset et d'enrichir la diversité des échantillons d'eau représentés. Cette approche de data augmentation vise à améliorer la capacité de notre modèle à généraliser et à prédire avec précision les classes de turbidité des échantillons d'eau.

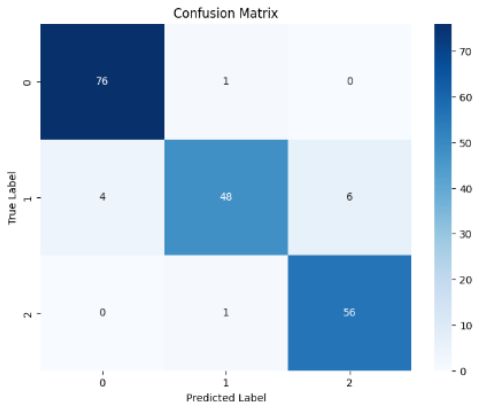
En total après expansion de l'ancien dataset puis en appliquant les méthodes de variation de texture et de contraste on un obtenu un dataset avec 960 images répartis comme présenté dans la figure 3.22 .

CLASSE	NOMBRE D'IMAGES
Turbidité Faible	352
Turbidité Moyenne	283
Turbidité élevée	325

FIGURE 3.22 – Répartition du dataset après l'augmentation des données.

Lors de l'utilisation des modèles, nous avons réservé 20% des images du dataset pour la validation, ce qui correspond à 192 images. Les 80% restants, soit 768 images, ont été utilisés pour l'entraînement des modèles. Cette séparation entre les données d'entraînement et de validation nous permet d'évaluer les performances des modèles sur des données non vues auparavant et d'obtenir une estimation réaliste de leur précision

Nous avons entraîné un modèle Random Forest en utilisant uniquement les caractéristiques qui ne contiennent pas de valeurs manquantes. Les résultats obtenus sont présentés dans la figure 3.23.



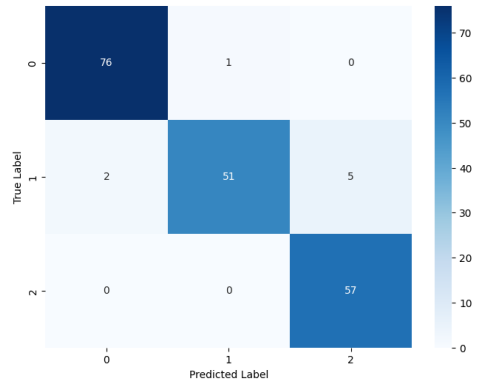
Matrice de confusion

CLASSE	PRÉCISION	RAPPEL	F1-SCORE	SUPPORT
Turbidité Faible	0.95	0.99	0.97	77
Turbidité Moyenne	0.96	0.83	0.89	58
Turbidité élevée	0.9	0.98	0.94	57
		Macro AVG	0.93	192
		Weighted AVG	0.94	192
			Accuracy	0.94

Rapport

FIGURE 3.23 – Matrice de confusion et Rapport du modèle Random Forest utilisant uniquement les caractéristiques sans valeurs manquantes.

Nous avons entraîné un modèle Random Forest en utilisant les 20 meilleures caractéristiques avec la méthode RFE.Les meilleures caractéristiques sont principalement liées à la texture, notamment les caractéristiques LBP (Local Binary Patterns) et Haralick, ainsi qu'aux caractéristiques de couleur. Les résultats obtenus sont présentés dans la figure 3.24.



Matrice de confusion

CLASSE	PRÉCISION	RAPPEL	F1-SCORE	SUPPORT
Turbidité Faible	0.97	0.99	0.98	77
Turbidité Moyenne	0.98	0.88	0.93	58
Turbidité élevée	0.92	1	0.96	57
		Macro AVG	0.96	192
		Weighted AVG	0.96	192
			Accuracy	0.96

Rapport

FIGURE 3.24 – Matrice de confusion et Rapport du modèle random forest utilisant RFE pour les 20 meilleures caractéristiques.

Comme indiqué dans les figures, l'augmentation des données a un impact très important sur les performances. Nous avons observé une amélioration significative de l'exactitude (accuracy) qui atteint maintenant 0.96. De plus, la prédiction de la classe 1 a été grandement améliorée, en particulier avec le modèle Random Forest utilisant la sélection des 20 meilleures caractéristiques avec RFE.

3.5 Conclusion

Le processus de travail consistait à entraîner différents modèles sur un dataset initial. Ensuite, dans le but d'améliorer les performances, nous avons décidé d'augmenter la taille du dataset en ajoutant 109 nouvelles photos, principalement des exemples de

classes de faible et moyenne turbidité. Cependant, malgré cette augmentation du dataset, les résultats obtenus avec les nouveaux modèles n'ont pas été meilleurs que ceux du modèle initial. Malgré nos efforts pour rendre le modèle plus généralisé en augmentant la diversité des données, plusieurs facteurs ont contribué à des performances moins bonnes. Premièrement, la taille du dataset initial était assez petite, ce qui limitait la capacité du modèle à généraliser. Deuxièmement, les valeurs de turbidité fournies par le capteur étaient imprécises, introduisant du bruit dans les données et rendant les prédictions moins précises.

Ensuite, dans le processus de data augmentation de notre ensemble de données, nous avons utilisé deux méthodes pour varier la texture et le contraste des images. Nous avons obtenu un large dataset équilibré et les résultats se sont considérablement améliorés, atteignant une exactitude (accuracy) de 0.96.

En résumé, le processus de travail consistait à entraîner des modèles sur un dataset initial, puis à augmenter la taille du dataset pour améliorer les performances. ce qui n'a pas montré de meilleurs résultats .Avec la partie de l'augmentation des données nous avons obtenu des résultats meilleurs

Conclusion

En conclusion, notre projet s'est concentré sur le développement d'une application mobile pour l'analyse de l'eau et la prédiction des niveaux de turbidité de l'eau à l'aide de diverses techniques de traitement d'image et d'apprentissage automatique. Grâce à notre travail, nous avons démontré le potentiel de ces méthodes pour évaluer avec précision la qualité de l'eau.

Bien que nos modèles d'apprentissage automatique aient montré des résultats prometteurs dans la prédiction de la turbidité sur la base des caractéristiques de l'image, nous reconnaissons que des améliorations supplémentaires peuvent être apportées. L'exploration des modèles d'apprentissage profond est une voie possible d'amélioration. Les techniques d'apprentissage en profondeur, telles que les réseaux neuronaux convolutifs (CNN), ont démontré des performances supérieures dans les tâches d'analyse d'images. En tirant parti de la puissance de l'apprentissage profond, nous pouvons potentiellement atteindre une précision et une robustesse encore plus grandes dans nos prédictions.

En outre, les futures itérations de notre projet pourraient bénéficier de l'élargissement de la portée de la collecte et de l'analyse des données. L'augmentation de la diversité et de la quantité des échantillons d'eau, ainsi que la collecte de paramètres supplémentaires liés à la qualité de l'eau, pourraient permettre une compréhension plus complète des facteurs influençant la turbidité. Cet ensemble de données enrichi nous permettrait de construire des modèles plus robustes capables de saisir les relations complexes entre les caractéristiques de l'image et les niveaux de turbidité.

Dans l'ensemble, notre projet met en évidence le potentiel des techniques de traitement d'images et d'apprentissage automatique pour l'analyse de l'eau. En adoptant des modèles d'apprentissage profond et en incorporant l'augmentation des données, nous pouvons affiner notre approche et potentiellement obtenir des prédictions encore plus précises et plus fiables. Alors que nous continuons à faire progresser notre compréhension de l'évaluation de la qualité de l'eau, notre travail contribue à l'objectif plus large d'assurer des ressources en eau propres et sûres pour tous.

Bibliographie

[1] Jean-Michel Martinez *Mesure de la qualité des eaux par satellite.*

[2] Fredrik Bajers Vej,Aalborg University Department of Electronic Systems *Turbidity measurement based on computer vision* .

[3] Fredrik Bajers Vej,Aalborg University Department of Electronic Systems *Turbidity measurement based on computer vision* .

[4] Harsh Tiwari *Support Vector Machine(SVM) :I can do both classification and regression* .

[5] Tibco,reference-center *Qu'est-ce qu'une forêt aléatoire ?* .

[6] Abdullah Siddique *Exploring KNN with Different Distance Metrics*