



Hand-crafted visual features to predict Image Quality on eye fundus RGB images

Rapport de projet

Wassim Smati et Tristan Jin

Encadré par : Elsa Angelini

Table des matières

1	Introduction	2
1.1	Présentation du projet	2
1.2	Support et outils	2
1.3	Pipeline générale	2
2	Approche initiale : développement d'un modèle fonctionnel	3
2.1	Prétraitement	3
2.2	Choix des features	3
2.3	Apprentissage et résultats	4
2.3.1	Paramètres par défaut	4
2.3.2	Modification des paramètres du SVM	4
3	Tentatives d'amélioration	5
3.1	Prétraitement : égalisation d'histogramme locale (CLAHE)	5
3.2	Amélioration de l'extraction de features	6
3.2.1	Méthode des patchs	6
3.2.2	PyRadiomics	6
3.3	Augmentation artificielle du dataset	6
3.3.1	Augmentation par ajout de bruit gaussien	7
3.3.2	Augmentation par rotation des images	7
4	Éléments de cours appliqués et notions apprises	10
4.1	Traitemet d'images	10
4.1.1	Prétraitement	10
4.1.2	Extraction de features visuelles	10
4.2	Apprentissage automatique	11
4.2.1	Méthodes d'extraction de features	11
4.2.2	Gestion du déséquilibre de classes	11
4.2.3	Évaluation et exploitation des résultats	11
5	Conclusion	12

1. Introduction

1.1. Présentation du projet

Ce projet vise à résoudre un problème courant lors des campagnes de dépistage : les images du fond de l'œil (eye fundus) de mauvaise qualité, souvent mal capturées, qui deviennent inexploitables pour le diagnostic. L'objectif est de développer un modèle d'apprentissage capable de classifier automatiquement ces images selon leur qualité, permettant ainsi de filtrer celles qui sont inutilisables.

1.2. Support et outils

Nous nous appuyons sur trois articles de recherche, cités dans la bibliographie de ce rapport, qui abordent ce problème selon des approches différentes. Notre démarche consiste à développer notre propre modèle en intégrant les éléments les plus pertinents de chacune de ces études.

Concernant notre **dataset**, nous avons utilisé l'*Open ultrawidefield fundus image dataset* fourni par notre encadrante. Celui-ci comprend des images de fond d'œil réparties en **587 images de bonne qualité (classe 1) et 113 images de mauvaise qualité (classe 0)**. La distinction entre bonne et mauvaise qualité repose sur 4 critères : Field of View, Contrast, Artifacts, Illumination.

Deux points importants sont à souligner :

- **Taille limitée du dataset** : Le nombre total d'images reste relativement faible pour un apprentissage robuste. Par exemple, avec un échantillon de test de 20%, nous ne disposons que d'une vingtaine d'images de mauvaise qualité pour évaluer le modèle. Ainsi, une seule prédiction incorrecte peut entraîner des variations significatives de performance.
- **Déséquilibre des classes** : Le dataset présente une distribution fortement déséquilibrée, avec environ 80% d'images de bonne qualité contre 20% d'images de mauvaise qualité. Ce déséquilibre peut influencer le comportement du modèle, comme nous le verrons par la suite.

Concernant l'environnement de travail, le code est hébergé sur Google Colab tandis que le dataset est stocké sur Google Drive. Ce dernier comprend les images ainsi qu'un fichier Excel associant à chaque image son label de qualité.

1.3. Pipeline générale

À partir des images brutes du dataset, nous procédons en 6 étapes pour construire notre modèle :

1. **Importation du dataset et des bibliothèques** nécessaires (OpenCV, NumPy, Pandas, Scikit-learn, etc.).
2. **Prétraitement des images** : transformation des images afin de les rendre plus exploitables. Les images prétraitées sont ensuite sauvegardées sur le Drive.
3. **Extraction des features** : calcul des caractéristiques pertinentes pour chaque image.
4. **Construction des vecteurs X et Y** : création de la matrice de features **X** et du vecteur d'étiquettes **Y** à partir des images prétraitées, puis sauvegarde de l'ensemble dans un fichier CSV.
5. **Apprentissage du modèle** : entraînement d'un classifieur **SVM** sur les données extraites.
6. **Évaluation du modèle** : calcul de la **précision**, du **rappel**, du **F1-score** et de l'**AUC**, à la fois globalement et pour chaque feature prise individuellement.

2. Approche initiale : développement d'un modèle fonctionnel

2.1. Prétraitement

En nous inspirant des articles de recherche, nous avons établi le prétraitement suivant :

1. Redimensionnement des images à une résolution de 512×512 pixels
2. Extraction du canal vert
3. Égalisation d'histogramme
4. Suppression des bordures par rognage circulaire

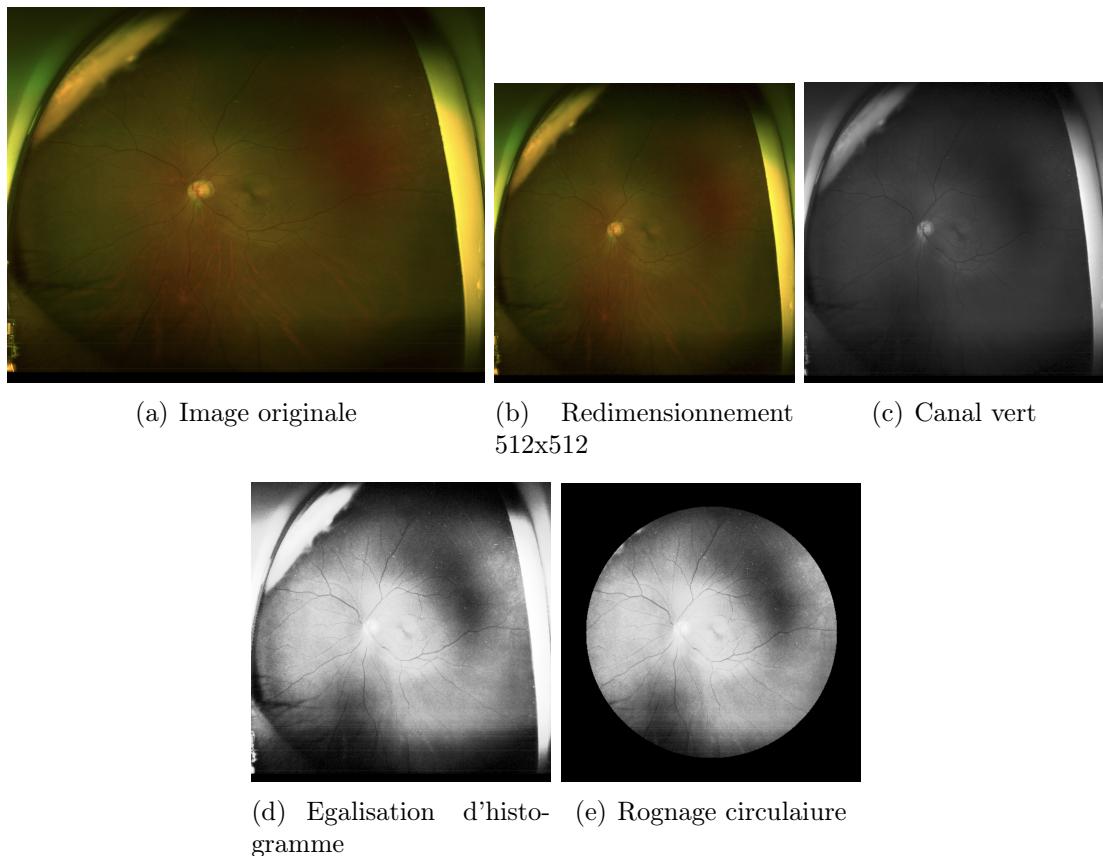


FIGURE 2.1 – Prétraitement naïf sur l'image de bonne qualité AMD-001

Remarque : Le choix du canal vert est expliqué dans la partie 4 du rapport.

2.2. Choix des features

Concernant le choix des features, nous avons sélectionné celles qui nous semblaient les plus pertinentes parmi celles utilisées dans les articles de référence.

1. **Features statistiques** : moyenne (mean), écart-type (std : standard deviation) et asymétrie (skewness).
2. **Features de texture** : descripteurs de Haralick (contraste, énergie, entropie).

3. **Features de contours** : netteté (sharpness) représentée par un couple de valeurs (s_1, s_2), où s_1 correspond à la *densité des contours forts* et s_2 à l'*intensité moyenne des contours*.

Notre vecteur de features est donc de dimension 8.

2.3. Apprentissage et résultats

2.3.1. Paramètres par défaut

Après avoir construit nos vecteurs de features X et de labels Y, nous entraînons notre modèle SVM sur 80% du dataset avec les paramètres par défaut :

— **kernel = RBF**

— **class_weight = None**

L'évaluation sur les 20% restants donne les résultats suivants :

Classe	Précision	Rappel	F1-Score	Support
0 (Mauvaise)	0.67	0.09	0.16	22
1 (Bonne)	0.85	0.99	0.92	118

TABLE 2.1 – RDF, unbalanced - **AUC = 0.541, Précision = 0.850**

2.3.2. Modification des paramètres du SVM

Nous constatons que les résultats obtenus avec les paramètres par défaut du SVM sont insatisfaisants. En particulier, la classe minoritaire (0) souffre d'un rappel très faible, indiquant que le modèle peine à détecter ses occurrences, ainsi que d'une précision faible, ce qui signifie que les rares prédictions effectuées pour cette classe sont majoritairement incorrectes.

Pour remédier à ce problème, nous avons modifié le paramètre **class_weight** en le réglant sur "**balanced**". Ce paramètre ajuste automatiquement les poids des classes, accordant ainsi davantage d'importance aux classes sous-représentées dans la fonction de coût du SVM. Voici les résultats obtenus après cette modification :

Classe	Précision	Rappel	F1-Score	Support
0 (Mauvaise)	0.42	0.68	0.52	22
1 (Bonne)	0.93	0.82	0.87	118

TABLE 2.2 – RBF, balanced - **AUC = 0.752, Précision = 0.800**

L'AUC devient nettement meilleur avec ce paramètre.

On peut encore modifier le paramètre **kernel**. On teste avec **kernel = linear**. Voici les résultats :

Classe	Précision	Rappel	F1-Score	Support
0 (Mauvaise)	0.43	0.68	0.53	22
1 (Bonne)	0.93	0.83	0.88	118

TABLE 2.3 – Linear, balanced - **AUC = 0.756, Précision = 0.807**

A la suite de ces différents tests, nous garderons les paramètres : Linear et Balanced pour la suite du projet.

3. Tentatives d'amélioration

Dans une première partie, nous avons implémenté la pipeline de manière simple, les seules améliorations apportées concernant les paramètres du modèle SVM. Après avoir ainsi optimisé le classifieur, nous allons désormais nous concentrer sur le traitement des images en amont.

3.1. Prétraitement : égalisation d'histogramme locale (CLAHE)

Notre encadrante nous a fait remarquer que notre prétraitement produisait une image finale surexposée, ce qui empêchait de visualiser correctement les vaisseaux sanguins présents sur l'image. La raison en est simple : l'égalisation d'histogramme est effectuée sur l'ensemble de l'image, qui contient des zones très claires non pertinentes pour notre modèle.

Pour résoudre ce problème, nous avons d'abord envisagé d'effectuer l'égalisation d'histogramme après le rognage, sans prendre en compte les bordures noires. Toutefois, nous avons finalement opté, sur conseil de notre encadrante, pour une autre méthode : **CLAHE (Contrast Limited Adaptive Histogram Equalization)**, qui réalise une égalisation d'histogramme localement sur de petites régions de l'image. Voici les résultats obtenus :

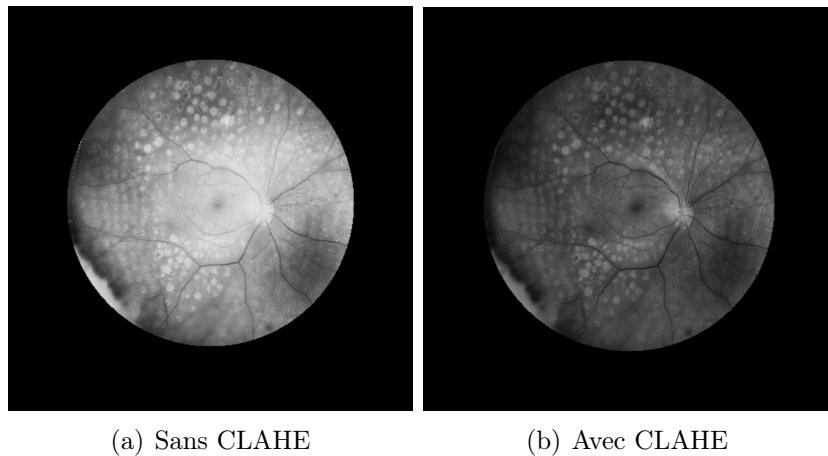


FIGURE 3.1 – Prétraitement de DR-012 avec et sans CLAHE

La différence entre les deux prétraitements est immédiate : grâce à l'égalisation CLAHE, l'image apparaît moins surexposée et devient ainsi nettement plus exploitable pour l'analyse.

Voici les résultats obtenus après entraînement du modèle SVM avec ce nouveau prétraitement :

Classe	Précision	Rappel	F1-Score	Support
0 (Mauvaise)	0.37	0.86	0.52	22
1 (Bonne)	0.97	0.73	0.83	118

TABLE 3.1 – Linear, Balance, Clahe - **AUC = 0.796, Précision = 0.750**

La précision de la classe 1 continue d'augmenter, tout comme l'AUC. En revanche, la précision de la classe 0 reste très faible en raison du déséquilibre du dataset et donc du nombre insuffisant d'images de cette classe pour l'apprentissage.

3.2. Amélioration de l'extraction de features

Maintenant que le modèle apprend correctement et dispose d'images mieux prétraitées, nous cherchons à lui fournir des informations plus précises.

3.2.1. Méthode des patchs

Afin d'enrichir notre représentation des images, nous avons implémenté la méthode des patchs. Cette approche consiste à découper une image en petites zones et à extraire des features sur chacune d'elles, permettant ainsi de capturer les détails locaux et d'obtenir une représentation spatiale plus fine de l'image.

Dans notre cas, nous avons découpé chaque image en 16 patchs carrés uniformes de taille 128×128 pixels. Cela multiplie par 16 le nombre de features, soit $16 \times 8 = 128$ features au total. Voici les résultats obtenus :

Classe	Précision	Rappel	F1-Score	Support
0 (Mauvaise)	0.55	0.95	0.70	22
1 (Bonne)	0.99	0.86	0.92	118

TABLE 3.2 – Linear, Balanced, Clahe, Patch - **AUC = 0.905, Précision = 0.871**

Cette méthode a permis d'améliorer significativement les performances du modèle : l'AUC atteint désormais 0.905 et la précision globale du modèle SVM progresse également. Nous conserverons donc cette méthode d'extraction de features pour la suite du projet.

3.2.2. PyRadiomics

PyRadiomics est une bibliothèque Python dédiée à l'extraction automatique de caractéristiques quantitatives à partir d'images médicales. Elle permet de calculer des centaines de features d'intensité, de texture et de forme sur des régions d'intérêt, offrant ainsi une représentation riche et détaillée des images. Voici les résultats obtenus avec cette approche :

Classe	Précision	Rappel	F1-Score	Support
0 (Mauvaise)	0.47	0.73	0.57	22
1 (Bonne)	0.94	0.85	0.89	118

TABLE 3.3 – Linear, Balanced, Clahe, PyRadiomics - **AUC = 0.787, Précision = 0.829**

Cette méthode n'apportant pas d'amélioration significative par rapport à notre modèle précédent, nous avons décidé de ne pas la retenir pour la suite.

Remarque : Nous n'avons pas tenté de combiner PyRadiomics avec la méthode des patchs, car cela aurait généré un nombre excessif de features, rendant l'apprentissage impraticable.

3.3. Augmentation artificielle du dataset

Dans cette partie, nous avons tenté de résoudre le problème de déséquilibre des classes dans notre dataset (classe 0 minoritaire). Rééquilibrer un dataset d'images est une tâche délicate, car chaque méthode visant à compenser le déséquilibre peut introduire des biais susceptibles de dégrader les performances du modèle ou de les améliorer de manière artificielle.

Nous nous sommes concentrées uniquement sur la méthode de *Data Augmentation*, qui consiste à augmenter artificiellement les données. Nous avons écarté l'oversampling (duplication des images

de la classe minoritaire) et l'undersampling (réduction de la classe majoritaire), car ces approches entraînent respectivement un risque de surapprentissage ou une perte d'informations importante.

3.3.1. Augmentation par ajout de bruit gaussien

L'objectif de cette approche était de générer de nouvelles images de classe 0 (mauvaise qualité) en ajoutant du bruit gaussien à des images de classe 1 (bonne qualité). Cependant, les résultats obtenus sur ces données synthétiques se sont révélés décevants, dégradant les performances du modèle plutôt que de les améliorer. Comme notre encadrante l'avait anticipé, le modèle a appris à détecter le bruit artificiel ajouté plutôt qu'à identifier véritablement les caractéristiques intrinsèques des images de mauvaise qualité naturelles. Ce phénomène s'apparente à un problème de *data shift* : les images synthétiquement bruitées ne suivent pas la même distribution que les véritables images de mauvaise qualité du jeu de données réel. Le modèle apprend donc à reconnaître un artefact de la méthode d'augmentation plutôt qu'à généraliser sur les vraies dégradations présentes dans les données d'origine.

3.3.2. Augmentation par rotation des images

Sensibilité des features aux rotations

Notre encadrante nous a demandé de vérifier si les features étaient sensibles aux rotations. Un premier test sur les features (mean, std, skew, sharpness, haralick) s'est révélé peu concluant, car ces dernières ne variaient pas pour des rotations de 90°, 180° et 270°.

Par la suite, elle nous a demandé de vérifier l'effet de la rotation sur **les features de texture extraites de la matrice de co-occurrence (GLCM)**. Ces features comprennent : **le contraste, l'énergie, la corrélation et l'homogénéité**, calculées dans **4 directions différentes (0°, 45°, 90°, 135°)**, soit **16 features au total**. Après avoir testé des rotations d'angles 15°, 30°, 45° et 90°, nous avons constaté que ces features variaient effectivement.

Cette analyse nous a permis d'identifier deux erreurs dans notre approche initiale :

1. En examinant plus en détail notre feature **Haralick**, composée notamment du contraste et de l'énergie, nous avons constaté qu'elle correspondait à la moyenne des valeurs calculées dans les quatre directions. Cela explique pourquoi nous observions presque aucune différence entre les descripteurs Haralick d'une image originale et de cette même image après rotation.
2. Nous avions uniquement testé des rotations d'angles multiples de 90°, et non des angles intermédiaires compris entre 0° et 90°.

Voici les résultats obtenus :

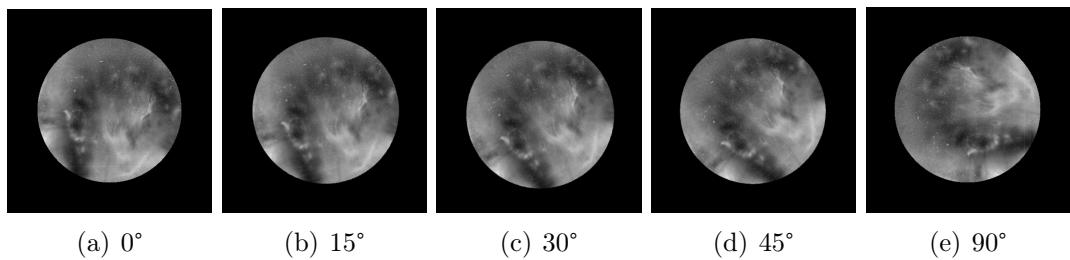


FIGURE 3.2 – Rotation de l'image AMD-005 (classe 0)

Features extraits de la GLCM :

- **0°** : [39.59, 47.24, 35.47, 53.32...]
- **15°** : [19.63, 28.07, 18.93, 33.43...]

- **30°** : [18.56, 28.19, 19.73, 32.98...]
- **45°** : [18.13, 29.38, 20.77, 32.41...]
- **90°** : [35.47, 53.32, 39.59, 47.24...]

Nouvelle fonction d'extraction de features

À partir de ce constat, nous avons développé une stratégie d'augmentation basée sur des rotations d'images. L'idée consiste à créer de nouvelles images de classe 0 en appliquant des rotations de 15°, 30° et 45° aux images originales, permettant ainsi de quadrupler le nombre d'exemples de cette classe minoritaire.

Toutefois, les features extraits initialement (notamment via le descripteur de Haralick) sont invariantes aux rotations. Appliquer de simples rotations aux images reviendrait donc à du sur-échantillonnage redondant sans apport d'information nouvelle. Il est donc nécessaire de modifier notre fonction d'extraction de features pour qu'elle soit sensible à l'orientation.

Nous avons choisi de remplacer le descripteur de **Haralick** par les features extraits directement de la matrice de co-occurrence de niveaux de gris (GLCM) : **4 features de contraste, 4 d'énergie, 4 de corrélation et 4 d'homogénéité**, correspondant aux 4 directions d'analyse (0°, 45°, 90°, 135°). Cela porte le nombre total de features à 21.

Ce remplacement est justifié par le fait que le descripteur de Haralick est lui-même dérivé de la GLCM, mais calcule une moyenne sur les 4 directions, entraînant une perte d'information directionnelle. En utilisant directement les valeurs de la GLCM pour chaque direction plutôt que leur moyenne, nous préservons davantage d'informations structurelles et texturales de l'image, notamment celles liées à l'orientation des motifs.

Résultats

Afin d'assurer la comparabilité des résultats avec les expériences précédentes et d'évaluer correctement la capacité de généralisation du modèle, nous avons appliqué un protocole expérimental rigoureux :

1. Le set d'entraînement comprend 80% des images originales de classe 0 ainsi que toutes les versions augmentées par rotation (générées à partir de l'ensemble du jeu de données de classe 0) ;
2. Le set de test reste strictement identique à celui utilisé dans toutes nos expériences précédentes : 20% des images originales de classe 0, sans aucune image augmentée. Cette cohérence garantit une évaluation comparable et fiable de l'apport réel de notre nouvelle méthode d'extraction de features.

L'objectif de cette stratégie est double : d'une part, enrichir le set d'entraînement avec des images augmentées pour exploiter pleinement les features sensibles à l'orientation ; d'autre part, maintenir un set de test composé uniquement d'images réelles pour évaluer les performances du modèle dans des conditions identiques aux expériences antérieures.

En combinant l'augmentation par rotation avec la nouvelle méthode d'extraction de features basée sur la GLCM directionnelle, nous obtenons les résultats suivants :

Analyse :

Ceci représente le meilleur résultat obtenu jusqu'à présent. Le modèle se montre particulièrement performant pour reconnaître les images de bonne qualité (classe 1), avec un F1-score de 0.94. Ce résultat est parfaitement cohérent avec notre objectif principal, qui était de développer un modèle capable d'identifier de manière fiable les images de bonne qualité.

Classe	Précision	Rappel	F1-Score	Support
0 (Mauvaise)	0.65	0.95	0.77	21
1 (Bonne)	0.99	0.90	0.94	114

TABLE 3.4 – Linear, Balanced, Clahe, Patch, DataAug par rotation - **AUC = 0.928, Précision = 0.911**

On observe également une amélioration sur la classe 0 : la précision passe de 0.55 (CLAHE + Patch) à 0.65. Même si cette progression reste modérée, elle n'a qu'un impact limité sur notre objectif final, dans la mesure où l'essentiel du modèle est justement d'être très efficace sur la détection des images de bonne qualité.

4. Éléments de cours appliqués et notions apprises

4.1. Traitement d'images

4.1.1. Prétraitement

Notions de cours appliquées

- **Canaux RGB** : Dans notre projet, nous avons choisi d'extraire le canal vert des images car sa longueur d'onde maximise le contraste entre le sang rouge des vaisseaux sanguins et le tissu rétinien environnant.
- **Égalisation d'histogramme** : Cette technique permet d'améliorer le contraste d'une image en redistribuant de manière plus uniforme les intensités des pixels sur l'ensemble de la plage dynamique disponible.

Notions nouvelles découvertes

- **CLAHE (Contrast Limited Adaptive Histogram Equalization)** : Contrairement à l'égalisation d'histogramme globale, CLAHE divise l'image en petites régions locales (tuiles) et applique une égalisation de l'histogramme à chacune d'elles. Cette approche locale préserve mieux les détails fins et évite l'amplification excessive du bruit.
- **Rognage (cropping)** : Technique permettant d'éliminer les zones non informatives en périphérie des images rétiniennes (zones noires dues au champ circulaire de capture).

4.1.2. Extraction de features visuelles

Notions de cours appliquées

- **Features statistiques de premier ordre** :
 - **Luminosité moyenne** (mean) : mesure l'intensité lumineuse globale de l'image
 - **Écart-type** (std) : quantifie la dispersion des intensités de pixels autour de la moyenne
- **Détection de contours - Filtre de Sobel** : Pour calculer la netteté (sharpness), nous avons utilisé les filtres de Sobel horizontal (S_x) et vertical (S_y) afin d'estimer la norme du gradient :

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

La norme du gradient permet de quantifier l'intensité des variations locales d'intensité, indicateur de la netteté de l'image.

- **Matrice de co-occurrence de niveaux de gris (GLCM)** : Cette matrice capture les relations spatiales entre pixels voisins et permet d'extraire des features de texture telles que :
 - **Contraste** : mesure les variations locales d'intensité
 - **Énergie** : quantifie l'uniformité de la texture
 - **Homogénéité** : évalue la proximité de la distribution dans la GLCM par rapport à sa diagonale
 - **Corrélation** : mesure la dépendance linéaire entre pixels voisins

Notions nouvelles découvertes

- **Skewness (asymétrie)** : Cette statistique de troisième ordre décrit l'asymétrie de la distribution des intensités de pixels par rapport à leur moyenne. Elle indique si les pixels sont davantage concentrés vers des valeurs faibles (asymétrie positive) ou élevées (asymétrie négative).

tive) :

$$\text{skewness}(I) = \frac{1}{N} \sum_{i=1}^N \left(\frac{I_i - \mu}{\sigma} \right)^3$$

où I_i représente l'intensité du pixel i , μ la moyenne, σ l'écart-type, et N le nombre total de pixels.

4.2. Apprentissage automatique

4.2.1. Méthodes d'extraction de features

Notions nouvelles découvertes

- **Méthode des patchs** : Cette approche consiste à découper une image en plusieurs sous-régions (patchs) pour extraire des features locales de manière indépendante, permettant ainsi de multiplier le nombre d'échantillons tout en capturant les variations spatiales au sein de l'image.
- **PyRadiomics** : Découverte et utilisation de cette bibliothèque Python spécialisée dans l'extraction de features radiomiques pour l'analyse d'images médicales. Elle offre un large éventail de descripteurs standardisés (forme, intensité, texture).

4.2.2. Gestion du déséquilibre de classes

Notions nouvelles découvertes

- **Augmentation de données (Data Augmentation)** : Technique consistant à créer artificiellement de nouveaux exemples de la classe minoritaire par transformations (rotations, translations, ajouts de bruit, etc.). Cette méthode requiert une attention particulière pour éviter le sur-apprentissage (*overfitting*), notamment en s'assurant que les transformations appliquées préservent les caractéristiques réelles de la classe et n'introduisent pas de biais artificiels détectables par le modèle.
- **Risque de data shift** : Prise de conscience que des données synthétiques peuvent suivre une distribution différente des données réelles, conduisant le modèle à apprendre des artefacts de la méthode d'augmentation plutôt que les véritables caractéristiques discriminantes.

4.2.3. Évaluation et exploitation des résultats

Notions nouvelles découvertes

- **Métriques d'évaluation** : Compréhension approfondie des différentes métriques de performance :
 - **Précision** : proportion de prédictions positives correctes
 - **Rappel** : proportion d'exemples positifs correctement identifiés
 - **F1-score** : moyenne harmonique de la précision et du rappel
 - **AUC (Area Under the Curve)** : aire sous la courbe ROC, mesurant la capacité du modèle à discriminer entre classes
- **Complexité de l'interprétation** : Prise de conscience de la difficulté à interpréter et exploiter les performances d'un modèle, notamment dans le choix de la métrique la plus pertinente selon le contexte applicatif et les coûts relatifs des différents types d'erreurs (faux positifs vs. faux négatifs).

5. Conclusion

Ce projet nous a permis de développer un modèle de classification automatique de la qualité d'images de fond d'œil. En partant d'une approche initiale simple inspirée des articles de référence, nous avons progressivement amélioré les performances du modèle à travers une démarche méthodique et itérative.

Les résultats obtenus sont encourageants : notre meilleur modèle atteint un AUC de 0.928 et une précision globale de 91.1%, avec un F1-score de 0.94 sur la classe 1 (bonne qualité). Ces performances démontrent la capacité du modèle à identifier de manière fiable les images exploitables pour le diagnostic, ce qui constitue l'objectif principal du projet.

Plusieurs facteurs ont contribué à ces résultats :

- L'amélioration du prétraitement avec l'utilisation de CLAHE, qui a permis d'obtenir des images mieux contrastées et plus exploitables
- L'adoption de la méthode des patchs, qui a significativement enrichi la représentation spatiale des images
- Le remplacement des features de Haralick par les features directionnelles de la GLCM, offrant une sensibilité aux orientations
- L'augmentation du dataset par rotation, permettant de mieux équilibrer les classes tout en respectant la distribution réelle des données

Toutefois, le déséquilibre initial du dataset (80% classe 1, 20% classe 0) et sa taille relativement modeste (700 images) ont représenté des contraintes importantes tout au long du projet. Malgré nos efforts d'augmentation de données, la précision sur la classe 0 reste modérée (0.65), même si le rappel élevé (0.95) garantit que peu d'images de mauvaise qualité passent inaperçues.

Ce projet nous a également permis d'approfondir nos connaissances en traitement d'images médicales et en apprentissage automatique. Nous avons notamment pris conscience de l'importance du choix des features, des risques liés à l'augmentation artificielle de données (data shift), et de la complexité de l'évaluation des performances d'un modèle dans un contexte applicatif réel.

Pour aller plus loin, plusieurs pistes d'amélioration pourraient être explorées : l'acquisition de davantage d'images de mauvaise qualité pour enrichir le dataset, l'expérimentation avec d'autres techniques d'augmentation de données plus sophistiquées, ou encore l'exploration d'approches basées sur l'apprentissage profond qui pourraient automatiquement apprendre des représentations plus riches des images.

Bibliographie

- [1] Yao Z, Zhang Z, Xu LQ, Fan Q, Xu L. *Generic features for fundus image quality evaluation.* IEEE International Conference on e-Health Networking, Applications and Services (Healthcom). 2016.
- [2] Wang S, Jin K, Lu H, Cheng C, Ye J, Qian D. *Human visual system-based fundus image quality assessment of portable fundus camera photographs.* IEEE Transactions on Medical Imaging. 2015.
- [3] Paulus J, Meier J, Bock R, Hornegger J, Michelson G. *Automated quality assessment of retinal fundus photos.* International Journal of Computer Assisted Radiology and Surgery, 2010.