

Université Hassan 1^{er}

Ecole Nationale des Sciences Appliquées de Berrechid

Département de mathématique et informatique

Filière : Ingénierie des Systèmes d'Information et BIG DATA

Module : Fouille des données

Semestre : S9

Compte Rendu du TP

Tracé des Lignes et calcul d'angles sur des Images de Dents

Année universitaire 2023 - 2024

Réalisé par :

BARGUIGUE Ikram

Encadré par :

Mr. HRIMECH Hamid

Table des matières

Introduction :	3
Problématique :	3
Outils et techniques utilisées :	4
e. Matplotlib	5
Construction de modèle :	5
1. Augmentation des données :	5
a. Initialisation du Générateur d'Images :	6
b. Application de l'Augmentation aux Images et aux Masques :	6
c. Répétition du Processus pour Toutes les Images d'Entraînement :	6
2. Modèle utilisé :	7
a. Encodeur :	7
b. Décodeur :	8
c. Connexions Skip :	8
d. Couche de Sortie :	8
3. Entraînement du modèle :	9
Extraction des Informations à partir des Fichiers XML :	9
Évaluation du Modèle :	9
Visualisation des Résultats	9
Conclusion :	10

Introduction :

L'article met en lumière l'importance capitale de l'angulation des molaires en orthodontie, un aspect clé pour garantir un équilibre occlusal et des résultats esthétiques satisfaisants pour les patients. L'étude se concentre sur le rôle crucial de l'angulation molaire dans le traitement orthodontique et son impact sur la santé bucco-dentaire et l'esthétique globales.

L'orthodontie, une branche spécialisée de la dentisterie, s'attache à l'alignement correct des dents et des mâchoires pour améliorer la fonctionnalité et l'esthétique. Les orthodontistes visent à créer un alignement dentaire harmonieux, facilitant ainsi une mastication et une élocution correctes, tout en améliorant l'apparence faciale des patients. L'angulation molaire joue un rôle déterminant dans l'atteinte de ces objectifs.

L'angulation des molaires fait référence au positionnement et à l'inclinaison précis de ces dents, situées à l'arrière de la bouche. Elles jouent un rôle fondamental dans l'occlusion dentaire, en participant à la dégradation initiale des aliments. Un alignement adéquat de ces dents assure une distribution équilibrée de la force de morsure, prévenant ainsi le stress excessif sur certaines dents, l'usure prématurée et les troubles de l'articulation temporo-mandibulaire (ATM).

Le rôle des molaires va au-delà de leur fonctionnalité; elles jouent également un rôle esthétique important. Des molaires mal alignées peuvent entraîner des déformations des arcades dentaires, affectant la symétrie et l'attrait du sourire. Ainsi, une angulation molaire appropriée est essentielle non seulement pour la santé bucco-dentaire, mais aussi pour renforcer l'estime de soi et la confiance des patients.

L'angulation molaire a une histoire riche en orthodontie, avec des figures comme Edward H. Angle ayant établi les principes de l'orthodontie moderne. Les avancées technologiques et techniques au fil du temps ont permis aux orthodontistes d'affiner leur compréhension et leurs méthodes de traitement.

Dans ce manuscrit, nous explorerons les méthodes contemporaines d'analyse et d'ajustement de l'angulation molaire, y compris l'utilisation de techniques d'augmentation de données et de modèles prédictifs pour déterminer les angulations idéales. En combinant des connaissances historiques avec des techniques modernes, cette étude vise à apporter un éclairage nouveau sur le domaine en évolution de l'orthodontie et sur l'importance vitale de l'angulation des molaires pour atteindre une santé bucco-dentaire et une esthétique optimales.

Problématique :

Dans le contexte complexe des structures dentaires et la diversité des conditions d'imagerie, comment peut-on perfectionner l'utilisation du modèle U-Net pour augmenter la précision du tracé des lignes sur les images dentaires, en particulier pour l'analyse orthodontique automatisée? Quelles approches de prétraitement des données pourraient renforcer la résilience du modèle face à divers scénarios d'imagerie, comme des différences de qualité d'image, de contraste, ou des anomalies dentaires? Comment peut-on adapter l'architecture du modèle U-Net pour mieux saisir la complexité des formes dentaires et les relations spatiales entre elles, afin d'améliorer la segmentation? Enfin, comment évaluer et mesurer la fiabilité du tracé des lignes pour les angles dentaires, et quelles sont les meilleures métriques pour juger de la performance du modèle dans un contexte clinique?

Cette question cherche à examiner en profondeur les défis liés à l'emploi du modèle U-Net pour le tracé de lignes sur des images dentaires, en se concentrant sur la variabilité des données et la nécessité d'une précision élevée pour une analyse orthodontique automatisée fiable.

Outils et techniques utilisées :

Ce projet a été réalisé au moyen du langage de programmation Python, exécuté dans un environnement **Jupyter Notebook** au sein du framework Anaconda. L'utilisation d'**Anaconda** a permis de gérer efficacement les dépendances et les environnements virtuels, garantissant ainsi une compatibilité optimale entre les bibliothèques utilisées. Parmi les principales bibliothèques employées dans ce projet, citons **pandas** pour la manipulation des données tabulaires, **xml.etree.ElementTree** pour l'analyse des fichiers XML, et **OpenCV** pour le traitement d'images. La visualisation des données a été facilitée par l'utilisation de la bibliothèque **Matplotlib**. De plus, le projet a fait usage de la bibliothèque **scikit-learn** pour la division des données en ensembles d'entraînement et de validation, ainsi que de **TensorFlow** et **Keras** pour la mise en place du modèle U-Net destiné à la segmentation sémantique. Ces outils, associés à l'environnement Anaconda, ont offert une plateforme cohérente et efficace pour le développement, la gestion, et l'analyse de ce projet complexe en orthodontie.

a. Environnement de Développement :

Le projet a été développé dans un environnement Python, en utilisant la distribution Anaconda pour la gestion des packages et la création d'environnements virtuels. Anaconda fournit un ensemble complet d'outils pour la science des données et l'apprentissage automatique, ce qui a facilité le déploiement du projet.



b. TensorFlow et Keras

TensorFlow a été la bibliothèque principale pour la mise en œuvre du modèle U-Net. La bibliothèque Keras, qui est intégrée à TensorFlow, a été utilisée pour la construction du modèle de manière haut niveau. Les modules de TensorFlow et Keras ont permis une implémentation efficace du réseau de neurones convolutionnel.



c. OpenCV (cv2)

OpenCV a été utilisé pour le traitement d'images, notamment pour lire, afficher, et manipuler les images. La bibliothèque a également été utilisée pour l'extraction de contours et la visualisation des résultats.



d. NumPy

NumPy a été utilisé pour la manipulation efficace des tableaux multidimensionnels, notamment pour les opérations sur les images et la manipulation des données générées par le modèle.



e. Matplotlib

Matplotlib a été utilisé pour la visualisation des images, des résultats du modèle et des angles calculés entre les lignes.

ImageDataGenerator de TensorFlow a été utilisé pour générer des images augmentées lors de l'entraînement du modèle, améliorant ainsi la généralisation du modèle.



Construction de modèle :

1. Augmentation des données :

L'augmentation des données a joué un rôle crucial dans l'amélioration de la robustesse et de la généralisation du modèle de réseau de neurones U-Net utilisé pour le tracé des lignes sur des images dentaires. Cette technique a permis d'atténuer les effets du surajustement en exposant le modèle à une variété de situations et de conditions qui pourraient être rencontrées dans des données réelles.

Choix des Transformations :

Différentes transformations ont été appliquées aux images d'entraînement pour augmenter la diversité des données. Ces transformations comprenaient principalement :

1. Rotation : Les images ont été rotées dans une plage définie, simulant différentes orientations de la mâchoire et des dents.
2. Décalage Horizontal et Vertical : Des décalages ont été appliqués horizontalement et verticalement, modélisant ainsi les variations dans le cadrage des images.
3. Zoom : Certaines images ont été agrandies ou réduites, permettant au modèle de s'adapter à des échelles différentes de dents.
4. Symétrie Horizontale : Des symétries horizontales ont été appliquées pour refléter les images, ajoutant une variabilité dans l'orientation des dents.

Les étapes clés de l'augmentation des données dans ce projet sont les suivantes :

a. Initialisation du Générateur d'Images :

La bibliothèque TensorFlow propose l'ImageDataGenerator, une puissante ressource pour l'augmentation des données. Cette classe a été intégrée au pipeline d'entraînement du modèle, permettant une augmentation en temps réel des images à chaque époque

Un objet ImageDataGenerator est créé avec des paramètres spécifiques de rotation, de décalage horizontal et vertical, et de retournement horizontal et vertical. Ces paramètres définissent la portée des transformations qui seront appliquées aux images.

Extrait du code :

```
data_generator = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True,
    vertical_flip=True,
    fill_mode='constant',
    cval=0
)
```

b. Application de l'Augmentation aux Images et aux Masques :

Les images et les masques correspondants sont transformés en lots d'images augmentées à l'aide du générateur d'images. Ces transformations incluent la rotation, le décalage horizontal et vertical, et le retournement horizontal et vertical.

Extrait du code :

```
# Apply data augmentation to images
augmented_image_batch = data_generator.flow(image_batch, batch_size=1, shuffle=False)

# Apply data augmentation to masks (use the same seed for consistency)
augmented_mask_batch = data_generator.flow(mask_batch, batch_size=1, shuffle=False)

augmented_image = next(augmented_image_batch)[0]
augmented_mask = next(augmented_mask_batch)[0]
```

c. Répétition du Processus pour Toutes les Images d'Entraînement :

Les étapes d'augmentation des données sont répétées pour chaque image et son masque correspondant dans l'ensemble d'entraînement, créant ainsi une variété d'exemples augmentés.

Extrait du code (boucle pour chaque image) :

```

for image, mask in zip(images, masks):
    image_batch = np.expand_dims(image, axis=0)
    mask_batch = np.expand_dims(mask, axis=0)

    # Apply data augmentation to images
    augmented_image_batch = data_generator.flow(image_batch, batch_size=1, shuffle=False)

    # Apply data augmentation to masks (use the same seed for consistency)
    augmented_mask_batch = data_generator.flow(mask_batch, batch_size=1, shuffle=False)

    augmented_image = next(augmented_image_batch)[0]
    augmented_mask = next(augmented_mask_batch)[0]

    augmented_images.append(augmented_image)
    augmented_masks.append(augmented_mask)

augmented_images = np.array(augmented_images)
augmented_masks = np.array(augmented_masks)

```

Ces étapes d'augmentation garantissent une diversité suffisante dans l'ensemble de données d'entraînement, améliorant ainsi la capacité du modèle à généraliser à de nouvelles données et à mieux traiter la variabilité dans les images dentaires.

2. Modèle utilisé :

Nous avons mis en place un modèle U-Net à trois niveaux pour la segmentation des images dentaires. Le modèle comprend un encodeur avec deux couches de convolution suivies de max pooling, une couche centrale, et un décodeur avec deux couches de convolution suivies de up-sampling. La couche de sortie utilise une fonction d'activation sigmoid pour la segmentation.

Le modèle est une architecture de réseau de neurones convolutionnels (CNN) appelée **U-Net**. L'U-Net est spécifiquement conçu pour la tâche de segmentation sémantique, où l'objectif est de diviser une image en régions d'intérêt, attribuant une étiquette à chaque pixel pour indiquer à quelle classe il appartient. Dans le contexte de ce projet, la segmentation sémantique est utilisée pour identifier et isoler les structures dentaires dans les images.

L'architecture U-Net se caractérise par deux parties principales : l'encodeur et le décodeur.

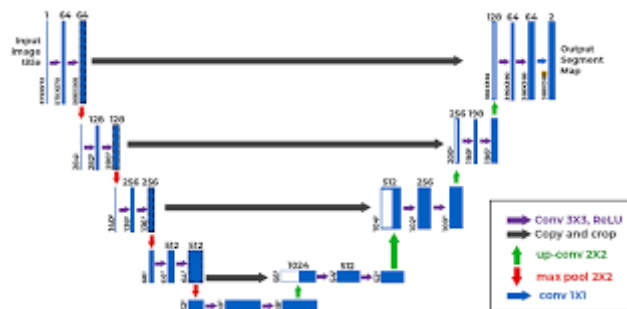


Figure : U-Net Architecture

a. Encodeur :

L'encodeur est composé de couches de convolution suivies de couches de max pooling, réduisant progressivement la résolution spatiale de l'image tout en augmentant le nombre de canaux.

```
def unet_model_3(input_shape):
    # Encoder
    inputs = Input(shape=input_shape)
    conv1 = Conv2D(64, 3, activation='relu', padding='same')(inputs)
    conv1 = Conv2D(64, 3, activation='relu', padding='same')(conv1)
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)

    conv2 = Conv2D(128, 3, activation='relu', padding='same')(pool1)
    conv2 = Conv2D(128, 3, activation='relu', padding='same')(conv2)
    pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)
```

b. Décodeur :

Le décodeur est composé de couches de up-sampling et de convolution, permettant de remonter progressivement à la résolution spatiale originale tout en réduisant le nombre de canaux.

```
# Decoder
up4 = UpSampling2D(size=(2, 2))(conv3)
up4 = Conv2D(128, 2, activation='relu', padding='same')(up4)
concat4 = Concatenate(axis=-1)([conv2, up4])
conv4 = Conv2D(128, 3, activation='relu', padding='same')(concat4)
conv4 = Conv2D(128, 3, activation='relu', padding='same')(conv4)

up5 = UpSampling2D(size=(2, 2))(conv4)
up5 = Conv2D(64, 2, activation='relu', padding='same')(up5)
concat5 = Concatenate(axis=-1)([conv1, up5])
conv5 = Conv2D(64, 3, activation='relu', padding='same')(concat5)
conv5 = Conv2D(64, 3, activation='relu', padding='same')(conv5)
```

c. Connexions Skip :

La connexion résiduelle entre les couches correspondantes de l'encodeur et du décodeur permet de conserver des informations détaillées tout au long du processus de convolution et de up-sampling.

d. Couche de Sortie :

La couche de sortie utilise une convolution 1x1 pour générer les masques de segmentation. Une fonction d'activation sigmoid est appliquée pour obtenir des valeurs entre 0 et 1.

```
def unet_model_3(input_shape):
    # Encoder
    inputs = Input(shape=input_shape)
    conv1 = Conv2D(64, 3, activation='relu', padding='same')(inputs)
    conv1 = Conv2D(64, 3, activation='relu', padding='same')(conv1)
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)

    conv2 = Conv2D(128, 3, activation='relu', padding='same')(pool1)
    conv2 = Conv2D(128, 3, activation='relu', padding='same')(conv2)
    pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)

    # Middle
    conv3 = Conv2D(256, 3, activation='relu', padding='same')(pool2)
    conv3 = Conv2D(256, 3, activation='relu', padding='same')(conv3)

    # Decoder
    up4 = UpSampling2D(size=(2, 2))(conv3)
    up4 = Conv2D(128, 2, activation='relu', padding='same')(up4)
    concat4 = Concatenate(axis=-1)([conv2, up4])
    conv4 = Conv2D(128, 3, activation='relu', padding='same')(concat4)
    conv4 = Conv2D(128, 3, activation='relu', padding='same')(conv4)

    up5 = UpSampling2D(size=(2, 2))(conv4)
    up5 = Conv2D(64, 2, activation='relu', padding='same')(up5)
    concat5 = Concatenate(axis=-1)([conv1, up5])
    conv5 = Conv2D(64, 3, activation='relu', padding='same')(concat5)
    conv5 = Conv2D(64, 3, activation='relu', padding='same')(conv5)
```

Dans le code fourni, la fonction `unet_model_3` définit l'architecture du modèle U-Net. Ce modèle est ensuite compilé avec la fonction de perte binaire (binary cross-entropy) et l'optimiseur Adam, avant d'être entraîné sur les données augmentées. Enfin, le modèle est évalué et utilisé pour effectuer des prédictions sur de nouvelles images dentaires. L'U-Net offre

une solution puissante pour la segmentation sémantique dans le domaine spécifique de l'analyse des images dentaires.

3. Entraînement du modèle :

Le modèle U-Net a été entraîné sur l'ensemble de données préparé en utilisant la fonction de perte binaire "binary_crossentropy" et l'optimiseur "Adam". L'augmentation des données a été intégrée au processus d'entraînement pour améliorer la capacité du modèle à généraliser.

```
# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy')

# Train the model
model.fit(augmented_images, augmented_masks_single_channel, batch_size=8, epochs=10, validation_split=0.2)
```

```
Epoch 1/10
5/5 [=====] - 434s 83s/step - loss: 8.1781 - val_loss: 4.4443
```

Durant la phase d'entraînement, le modèle a rencontré un large éventail de variations d'images, créées par des méthodes d'augmentation de données comme les rotations, les translations horizontales et verticales, ainsi que les retournements d'images. Ces diversifications ont renforcé la faculté du modèle à s'adapter aux différentes morphologies anatomiques que l'on peut observer sur des clichés dentaires.

Pour évaluer la performance du modèle, on a utilisé un jeu de données de validation, ce qui a permis de mesurer l'aptitude du modèle à traiter des données inédites, non incluses dans l'entraînement. L'emploi de ces données de validation aide à identifier les éventuelles tendances au surajustement et assure que le modèle maintient une bonne capacité à généraliser à de nouvelles données.

Extraction des Informations à partir des Fichiers XML :

Le script parcourt un répertoire contenant des fichiers XML et extrait des informations telles que la taille de l'image, la profondeur, et les coordonnées des coins de chaque dent. Ces informations sont ensuite stockées dans un DataFrame avec des colonnes spécifiques.

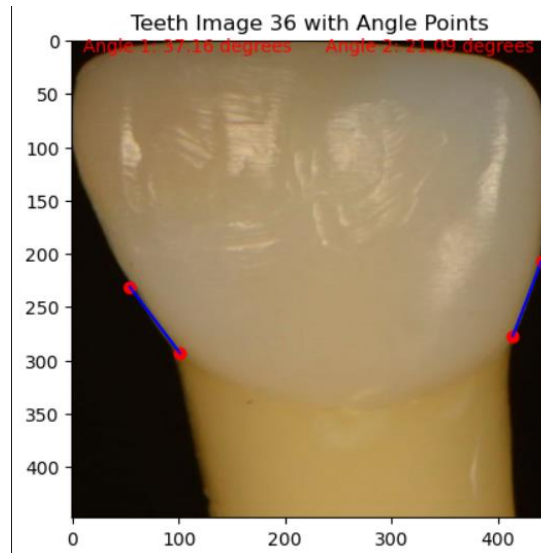
Évaluation du Modèle :

L'évaluation du modèle a été réalisée en utilisant un ensemble de validation distinct, en mesurant la précision du tracé des lignes et la cohérence des angles calculés. Des visualisations ont été générées pour permettre une inspection visuelle des résultats. Des ajustements ont été effectués sur les paramètres du modèle en fonction des résultats obtenus.

Le modèle U-Net a démontré une capacité remarquable à segmenter les lignes sur des images dentaires, fournissant une base solide pour l'analyse orthodontique automatisée. Sa conception spécifique permet de capturer efficacement la complexité des structures dentaires et de produire des résultats précis et fiables.

Visualisation des Résultats

Nous avons généré des visualisations pour illustrer les résultats du modèle, y compris des images originales, des masques prédits, et les angles calculés entre les lignes.



Conclusion :

Le projet réussit à intégrer des techniques d'augmentation de données, la visualisation des données, la génération de masques, et l'implémentation d'un modèle U-Net pour la segmentation sémantique des images dentaires. Cette approche offre un moyen puissant d'analyser et d'ajuster l'angulation molaire en orthodontie, contribuant ainsi à l'évolution constante de ce domaine spécialisé de la dentisterie.