

PROJET ANNUEL

NOM ET PRÉNOMS	PROJET
	01 Analyse en temps réel de flux vidéo : Détection, Segmentation et Comptage de véhicules
	02 Traduction EWE – Anglais, Anglais - EWE
	03 Entraîner votre propre mini modèle de génération de texte tel que ChatGPT
	04 Système de questions-réponses

Evaluation:

1. **Rapport:** Soumettre un rapport détaillant les étapes de réalisation, les difficultés, les solutions ainsi que les contributions de chaque member. A soumettre avant la date de presentation : tchaye59@gmail.com (10 points)
2. **Presentation:** Présentation des projets (10 points)

Date de Présentation: 31-août-24

LISTE DES PROJETS

I. PROJET 1 : ANALYSE EN TEMPS RÉEL DE FLUX VIDÉO : DÉTECTION, SEGMENTATION ET COMPTAGE DE VÉHICULES

Objectif : Analyser les objets dans une vidéo filmant une voie fréquentée par des véhicules segmentés et compter le nombre de véhicules sur chaque image. Lorsque l'utilisateur clique sur un véhicule, afficher sa plaque d'immatriculation.

Outils: Voici quelques outils et bibliothèques que vous pouvez utiliser pour réaliser votre projet:

1. OpenCV (Open Source Computer Vision Library) : OpenCV est une bibliothèque open-source très populaire pour le traitement d'images et la vision par ordinateur. Elle propose des fonctionnalités telles que la détection d'objets, la segmentation d'images et le suivi de mouvement. Vous pouvez l'utiliser pour extraire des informations à partir de vidéos.
2. YOLO (You Only Look Once) : YOLO est un modèle de détection d'objets en temps réel qui peut être utilisé pour détecter des véhicules dans des images ou des vidéos. Il est rapide et précis.
3. Mask R-CNN : Si vous avez besoin de segmenter les véhicules (c'est-à-dire les isoler du reste de l'image), Mask R-CNN est un excellent choix. Il permet la segmentation sémantique et peut être utilisé pour extraire des masques précis autour des véhicules.
4. NumPy et SciPy : Ces bibliothèques Python sont essentielles pour le traitement d'images, les calculs matriciels et les opérations statistiques. Vous pouvez les utiliser pour manipuler les données d'image et effectuer des calculs.
5. Tesseract OCR : Si vous souhaitez extraire des plaques d'immatriculation à partir d'images de véhicules, Tesseract OCR est un outil open-source qui peut reconnaître du texte dans des images.
6. Flask ou FastAPI : Pour créer une interface utilisateur et gérer les interactions avec l'utilisateur, vous pouvez utiliser Flask (pour une approche plus simple) ou FastAPI (pour une approche plus performante).

Étapes : Voici les étapes pour la réalisation de votre projet

1. **Collecte de données :**
 - Trouvez ou enregistrez des vidéos de voies fréquentées par des véhicules. Plus vous avez de données, mieux c'est pour l'entraînement de votre modèle.

2. Prétraitement des données :

- Utilisez des outils comme OpenCV pour extraire des images individuelles à partir de la vidéo.
- Redimensionnez les images pour une taille cohérente.
- Étiquetez manuellement les véhicules dans les images (si vous avez des annotations).

3. Détection d'objets :

- Entraînez un modèle de détection d'objets (comme YOLO) sur vos images annotées.
- Utilisez le modèle entraîné pour détecter les véhicules dans les images de la vidéo.

4. Segmentation d'images :

- Si vous souhaitez segmenter les véhicules, utilisez un modèle comme Mask R-CNN.
- Appliquez la segmentation aux véhicules détectés.

5. Comptage de véhicules :

- Pour chaque image, comptez le nombre de véhicules détectés.

6. Affichage des plaques d'immatriculation :

- Utilisez Tesseract OCR pour extraire le texte des plaques d'immatriculation.
- Lorsque l'utilisateur clique sur un véhicule détecté, affichez la plaque d'immatriculation correspondante.

7. Interface utilisateur :

- Créez une interface utilisateur (par exemple, avec Flask ou FastAPI) pour afficher les résultats et gérer les interactions.

8. Tests et optimisation :

- Testez votre modèle sur différentes vidéos.
- Optimisez les performances (par exemple, en ajustant les seuils de détection).

9. Déploiement :

- Déployez votre application pour qu'elle fonctionne en temps réel.

II. PROJET 2 : TRADUCTION EWE – ANGLAIS, ANGLAIS - EWE

Objectif : Créer un modèle de traduction bidirectionnelle entre l'EWE et l'anglais en utilisant le jeu de données fourni sur le lien suivant : [EWE-English Bilingual Pairs](#).

Etapes:

1. Prétraitement des données:

- Nettoyez et préparez vos données en supprimant les caractères spéciaux, les doublons et en normalisant le texte.
- Divisez vos données en ensembles d'entraînement, de validation et de test.

2. Choix du modèle :

- Créer votre propre modèle Transformateurs (comme BERT ou GPT). Les réseaux existents et préentraînés ne sont pas autorisés.
- Assurez-vous que votre modèle est capable de gérer les deux directions de traduction (EWE vers anglais et anglais vers EWE).

3. Entraînement du modèle :

- Alimenter votre modèle avec les paires de phrases EWE-anglais ([EWE-English Bilingual Pairs](#)).
- Expérimentez avec différentes architectures et hyperparamètres pour obtenir de bons résultats.

4. Évaluation du modèle :

- Évaluez les performances de votre modèle sur l'ensemble de test en utilisant des métriques telles que le BLEU score (qui mesure la qualité de la traduction).
- Réalisez des ajustements si nécessaire.

5. Inférence et déploiement :

- Utilisez votre modèle entraîné pour traduire des phrases de l'EWE vers l'anglais et vice versa.
- Vous pouvez déployer votre modèle en tant qu'API et l'intégrer dans une application (Web, mobile ou desktop).

III. PROJET 3 : ENTRAÎNER VOTRE PROPRE MINI MODÈLE DE GÉNÉRATION DE TEXTE TEL QUE CHATGPT

Objectif : Entraîner un mini assistant virtuel, similaire à ChatGPT (Mini ChatGPT), capable de répondre de manière humaine à chaque question. Utilisez les données d'OpenAssistant disponibles sur les liens suivants :

OpenAssistant/oasst1 (<https://huggingface.co/datasets/OpenAssistant/oasst1>) et **OpenAssistant/oasst2** (<https://huggingface.co/datasets/OpenAssistant/oasst2>). Concentrez-vous uniquement sur la langue française.

Etapes:

1. **Collecte de données :** Rassemblez un ensemble de données en français pour l'entraînement. Vous pouvez utiliser les données d'OpenAssistant (OpenAssistant/oasst1, OpenAssistant/oasst2).
2. **Prétraitement des données :** Nettoyez et prétraitez vos données. Cela peut inclure la suppression des balises HTML, la normalisation des textes, la tokenisation, etc.
3. **Créez votre modèle de transformateurs :** Concevez votre propre architecture de modèle de transformateurs. Vous pouvez utiliser des bibliothèques telles que Hugging Face Transformers
4. **Entraînement du modèle :** Entraînez votre modèle sur les données prétraitées. Utilisez des techniques telles que l'apprentissage supervisé.
5. **Déploiement :** Une fois que votre modèle est satisfaisant, déployez-le pour qu'il puisse répondre aux questions de manière humaine.

IV. PROJET 4 : SYSTÈME DE QUESTIONS-RÉPONSES : DÉVELOPPER UN SYSTÈME CAPABLE DE RÉPONDRE À DES QUESTIONS FORMULÉES EN LANGAGE NATUREL, BASÉ SUR UN ENSEMBLE DE DOCUMENTS DONNÉS.

Objectif : Développer un système intelligent capable de prendre des documents en entrée et de répondre à chaque question posée par l'utilisateur concernant ces documents. Utilisez des outils tels que Langchain.

Étapes

1. **Collecte de données** : Rassemblez un ensemble de documents pertinents qui serviront de base pour répondre aux questions. Ces documents peuvent être des articles, des livres, des rapports, etc.
2. **Prétraitement des données** : Nettoyez et formatez les documents pour faciliter leur analyse. Cela peut inclure la suppression des stop words, la lemmatisation, etc.
3. **Indexation des documents** : Créez un index des documents pour faciliter la recherche d'informations pertinentes lors de la réponse aux questions.
4. **Développement du modèle de questions-réponses** : Utilisez des techniques d'apprentissage automatique et de traitement du langage naturel pour développer un modèle capable de comprendre les questions en langage naturel et de rechercher des réponses dans l'ensemble de documents.
5. **Intégration de l'outil Langchain** : Intégrez l'outil Langchain pour améliorer les capacités de traitement du langage naturel du système.
6. **Test et évaluation** : Testez le système avec un ensemble de questions pour évaluer sa performance. Faites les ajustements nécessaires pour améliorer la précision des réponses.
7. **Déploiement** : Une fois que le système fonctionne de manière satisfaisante, déployez-le pour qu'il soit utilisé par les utilisateurs finaux.