

# **Document Technique de Configuration – Application IA Multimodale**

## **1. Informations générales**

Nom du projet : Application IA Multimodale

Auteur : Jane Grace davis Ngo Massing Soun

Modules : Analyse d'images (BLIP), Analyse de texte (NLP), Tableau de bord (Streamlit)

## **2. Architecture Logicielle**

L'application est organisée en trois modules principaux :

- Analyse d'Images : utilise le modèle BLIP pour générer une description textuelle.
- Analyse de Texte : utilise un pipeline NLP (sentiment-analysis).
- Tableau de Bord : zone prévue pour statistiques et visualisations futures.

L'interface web est gérée par Streamlit, tandis que l'IA repose sur HuggingFace Transformers.

## **3. Configuration de l'Environnement Virtuel**

1. Créer l'environnement virtuel :

```
python -m venv venv
```

2. Activer :

```
Windows : .\venv\Scripts\activate
```

3. Vérifier :

```
where python
```

```
where pip
```

## **4. Installation des Dépendances**

Dépendances nécessaires :

- pip install streamlit
- pip install pillow
- pip install transformers
- pip install timm
- pip install torch --index-url <https://download.pytorch.org/whl/cpu>

Ces bibliothèques permettent de faire fonctionner BLIP, les pipelines NLP et l'interface Streamlit.

## 6. Configuration du Module Image (BLIP)

Pipeline utilisé :

captioner = pipeline('image-to-text', model='Salesforce/blip-image-captioning-large')

Formats pris en charge : JPG, PNG, JPEG

Prétraitement effectué par PIL.

Sortie : description textuelle complète de l'image.

## 7. Configuration du Module Texte

Pipeline utilisé : sentiment-analysis

Entrée : texte libre

Sortie : label de sentiment + score.

Objectif : démontrer une analyse de texte simple et rapide.

## 8. Configuration de Streamlit

Lancement de l'application :

```
python -m streamlit run app.py
```

Port par défaut : 8501

Navigation : st.sidebar.radio permet de changer de module.

## 9. Débogage & Problèmes Courants

- Si Transformers ne charge pas : réinstaller transformers + timm.
- Si BLIP ne fonctionne pas : vérifier timm.
- Si Streamlit affiche l'ancienne version : CTRL+C, relancer.
- Si mauvais Python utilisé : vérifier where python.

## 10. Bonnes Pratiques

- Garder le venv isolé.
- Ajouter un .gitignore (ignorer venv/ et fichiers lourds).
- Ne pas stocker d'images sensibles.

## 11. Extensions Futures

- Ajouter une base de données SQLite.
- Ajouter un export PDF.
- Ajouter LLaVA comme modèle avancé.
- Ajouter des graphiques dans le tableau de bord.

## Captures de lexécution et du code

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL ... python + v [ ] [ ] ... | [ ] [ ] X

(venv) PS C:\Users\Jane Grace> python -m streamlit run app.py
Device set to use cpu
Stopping...
(venv) PS C:\Users\Jane Grace> python -m streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://172.16.25.218:8501

No model was supplied, defaulted to google/vit-base-patch16-224 and revision 3f49326
(https://huggingface.co/google/vit-base-patch16-224).
Using a pipeline without specifying a model name and revision in production is not recommended.
Device set to use cpu
No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-english and revision 714eb0f (https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english).
Using a pipeline without specifying a model name and revision in production is not recommended.
Device set to use cpu
No model was supplied, defaulted to google/vit-base-patch16-224 and revision 3f49326
(https://huggingface.co/google/vit-base-patch16-224).
Using a pipeline without specifying a model name and revision in production is not recommended.
Device set to use cpu
No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-english and revision 714eb0f (https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english).
Using a pipeline without specifying a model name and revision in production is not recommended.
Device set to use cpu
No model was supplied, defaulted to google/vit-base-patch16-224 and revision 3f49326
```

0 4 Ln 15, Col 1 Spaces: 4 UTF-8 CRLF { } Python 3.12.1

```

app.py 4 X
C:\Users> Jane Grace > app.py > ...
1 import streamlit as st
2 from transformers import pipeline
3 from PIL import Image
4 import pandas as pd
5 import os
6
7 st.set_page_config(page_title="Analyse IA Multimodale", layout="wide")
8
9 st.sidebar.title("Navigation")
10 page = st.sidebar.radio("Aller à :", ["Analyse d'Images", "Analyse de Texte", "Tableau de Bord"])
11
12 # Charger modèles IA
13 image_classifier = pipeline("image-classification")
14 sentiment_analyzer = pipeline("sentiment-analysis")
15
16 # Créer un historique s'il n'existe pas
17 if not os.path.exists("historique.csv"):
18     pd.DataFrame(columns=["type", "input", "result"]).to_csv("historique.csv", index=False)
19
20 historique = pd.read_csv("historique.csv")
21
22 #PAGE 1 : ANALYSE D'IMAGES
23 if page == "Analyse d'Images":
24     st.title("🖼️ Analyse d'Images avec IA")
25
26     uploaded_files = st.file_uploader("Choisis une ou plusieurs images", type=["jpg", "jpeg", "png"], accept_multiple_files=True)
27
28     if uploaded_files:
29         for uploaded_file in uploaded_files:
30             image = Image.open(uploaded_file)

```

```

Users / Jane Grace > app.py > ...
if page == "Analyse d'Images":
    st.title("🖼️ Analyse d'Images avec IA")

    uploaded_files = st.file_uploader("Choisis une ou plusieurs images", type=["jpg", "jpeg", "png"], accept_multiple_files=True)

    if uploaded_files:
        for uploaded_file in uploaded_files:
            image = Image.open(uploaded_file)
            st.image(image, caption=f"Image : {uploaded_file.name}", width=300)

            st.write("Analyse en cours...")
            results = image_classifier(image)

            st.write("Résultats :")
            for result in results:
                st.write(f"- {(result['label'])} ({(result['score']):.2f})")

            # sauvegarder dans historique
            historique.loc[len(historique)] = ["image", uploaded_file.name, results[0]["label"]]
            historique.to_csv("historique.csv", index=False)

# PAGE 2 : ANALYSE DE TEXTE
elif page == "Analyse de Texte":
    st.title("📝 Analyse de Texte avec IA")

    user_text = st.text_area("Écris un texte")

    if st.button("Analyser"):
        result = sentiment_analyzer(user_text)[0]
        st.write(f"Sentiment : {(result['label'])} (score : {(result['score']):.2f})")

```

```

        if st.button("Analyser"):
            result = sentiment_analyzer(user_text)[0]
            st.write(f"Sentiment : {(result['label'])} (score : {(result['score']):.2f})")

            historique.loc[len(historique)] = ["texte", user_text, result["label"]]
            historique.to_csv("historique.csv", index=False)

#PAGE 3 : TABLEAU DE BORD
else:
    st.title("📊 Tableau de bord des analyses")

    st.write("Voici l'historique d'utilisation :)")
    st.dataframe(historique)

    st.write("Graphique du nombre d'analyses par type :)")
    st.bar_chart(historique["type"].value_counts())

```