

## **École Marocaine des Sciences de l'Ingénieur – EMSI Tanger**

Filière : Ingénierie Informatique et Réseaux

Niveau : 5<sup>ème</sup> Année Cycle Ingénieur

### **Projets Ingénieur – ML & DL : Traitement du Langage Naturel (NLP) et Vision par Ordinateur**

Matière : ML & DL

Enseignant : Pr. Radouan DAHBI

Année universitaire : 2025–2026

## Contexte général

Dans le cadre de la matière Machine Learning & Deep Learning, vous réaliserez deux projets appliqués portant sur des problématiques réelles rencontrées dans l'industrie.

Ces projets sont pleinement alignés avec le syllabus du cours, couvrant notamment : Préparation et prétraitement des données, Feature Engineering et Data Augmentation, Modèles de Machine Learning supervisé et non supervisé, Deep Learning pour le Texte et Deep Learning pour les images.

Les projets proposés sont :

- **Projet 1 : Traitement du Langage Naturel (NLP)**  
Analyse de sentiments ou classification de textes.
- **Projet 2 : Vision par Ordinateur (Computer Vision)**  
Reconnaissance automatique d'expressions faciales.

Les deux projets suivent une structure similaire afin de vous permettre d'acquérir une méthodologie complète de développement d'un pipeline ML/DL, depuis l'exploration des données jusqu'à l'analyse des résultats.

## Organisation du travail

- **Travail en groupe** : équipes de 5 étudiants.
- **Durée estimée** : 6 à 7 semaines par projet, avec un travail réparti en séances guidées et un travail autonome.
- **Technologies recommandées** : Python, scikit-learn, TensorFlow/Keras ou PyTorch, Colab.

## Livrables finaux

- Code source complet (notebooks ou scripts) propre, bien organisé, commenté et reproductible.
- Rapport final (10 à 15 pages) présentant les choix techniques, les résultats, les limites du travail, les pistes d'amélioration ainsi que les références et les sources utilisées.
- Présentation orale (10-12 min par groupe) avec une répartition claire des contributions.

## Planning et livrables intermédiaires

Semaine	Travail attendu
Semaine 1	Constitution des groupes, choix du dataset, exploration initiale
Semaine 2	Prétraitement des données, préparation des features
Semaine 3	Implémentation des premiers modèles ML (baseline ML et MLP pour vision)
Semaine 4	Implémentation des premiers modèles DL (NLP : RNN, LSTM/GRU, Vision : CNN) et premières évaluations
Semaine 5	Transfer Learning, optimisation et tuning des hyperparamètres
Semaine 6	Clustering et analyse non supervisée
Semaine 7	Finalisation du rapport & préparation de la présentation

(Ce planning est flexible selon l'avancée de chaque groupe.)

#### Livrables attendus :

- Avancement intermédiaire (fin semaine 3) :
  - Format : code source initial, rapport court (3-4 pages)
  - Contenu : dataset choisi, exploration et prétraitement déjà réalisés, premiers modèles ML et premières métriques d'évaluation
- Prototype Deep Learning (fin semaine 5) :
  - Format : code source principal dédié au Deep Learning
  - Contenu : implémentation des premiers modèles DL, premiers résultats DL, mise en place du Transfer Learning, hyperparamètres testés, comparaison avec les baselines ML
- Livrables finaux (semaine 7) :
  - Format : rapport final, code source complet, propre et reproductible
  - Contenu : pipeline complet ML & DL, comparaison des modèles, clustering, analyse des résultats, difficultés rencontrées, pistes d'amélioration, conclusion et perspectives

#### Barème global d'évaluation

Critère d'évaluation	Description	Pondération	Points (/20)
Fonctionnalités & respect des consignes du projet	Mise en place d'un Pipeline complet : prétraitement, modèles ML & DL, transfer learning, clustering, résultats cohérents et reproductibles.	30 %	6 pts
Maîtrise des concepts ML/DL	Pertinence et justification des modèles, compréhension des métriques, tuning des hyperparamètres, analyse critique et comparaison des modèles.	25 %	5 pts
Qualité du code, structure & documentation	Code propre, clair, commenté, organisé et reproductible. Visualisations pertinentes (courbes, matrices de confusion, PCA/t-SNE).	20 %	4 pts
Travail d'équipe & gestion du projet	Répartition des tâches, régularité du travail, respect des jalons (S3/S5/S7), autonomie et organisation interne.	15 %	3 pts
Rapport final & soutenance orale	Rapport structuré (10–15 pages), clarté des analyses, limites & perspectives, qualité de la présentation orale, maîtrise technique.	10 %	2 pts

## PROJET 1 – Traitement du Langage Naturel (NLP)

**Durée :** 6 à 7 semaines

**Travail :** en groupes de 5 étudiants

### Objectif général :

Ce projet a pour objectif de simuler une tâche réelle de traitement automatique du langage naturel (NLP), telle qu'elle peut être rencontrée en entreprise, en choisissant l'un des deux axes suivants :

- **Analyse de sentiments** (positif / négatif / neutre ou sentiments détaillés)
- **Classification de textes** (classification thématique, catégorisation de documents, intentions utilisateur, types de commentaires, etc.)

Chaque groupe réalisera une analyse complète d'un corpus textuel volumineux, en expérimentant plusieurs approches de Machine Learning (ML) et de Deep Learning (DL) afin de comparer leurs performances.

Le projet comprend également :

- une gestion potentielle du déséquilibre des classes,
- une optimisation des hyperparamètres afin d'améliorer les performances des modèles,
- une analyse non supervisée pour explorer la structure thématique du corpus.

### 1. Description générale du projet :

Vous réaliserez une analyse complète sur un jeu de données spécifique. Le projet comprend les étapes suivantes :

1. Sélection du jeu de données (par groupe).
2. Exploration et prétraitement des données textuelles.
3. Mise en œuvre de modèles supervisés (ML).
4. Approche Deep Learning (DL).
5. Analyse non supervisée et Clustering.
6. Analyse du déséquilibre des classes et optimisation des modèles.
7. Rédaction du rapport final et présentation.

### 2. Jeux de données proposés :

Chaque groupe doit sélectionner un jeu de données pour son projet. Vous pouvez choisir l'un des jeux de données suggérés ci-dessous ou proposer un autre dataset équivalent, à condition qu'il réponde aux mêmes critères : un dataset volumineux et pertinent pour une tâche de traitement automatique du langage naturel (NLP).

Jeux de données proposés :

1. **Dataset 1 :** [IMDB Movie Reviews Dataset](#)  
Avis de films avec sentiments positifs et négatifs.
2. **Dataset 2 :** [Reddit Comments Sentiment](#)  
Commentaires Reddit annotés pour les sentiments.
3. **Dataset 3 :** [Reuters-21578](#)  
Articles de presse classés par thèmes économiques et financiers.
4. **Dataset 4 :** [Sentiment140](#)  
1,6 million de tweets annotés pour les sentiments.
5. **Dataset 5 :** [Wikipedia Toxic Comments](#)  
Commentaires issus de Wikipédia annotés pour différents types de toxicité.

**Remarque :** ces jeux de données sont disponibles sur Kaggle, assurez-vous de respecter les licences et conditions d'utilisation de chaque jeu de données avant de les exploiter. Les groupes souhaitant utiliser un autre dataset librement choisi doivent le faire valider par l'enseignant avant de commencer le projet.

### 3. Étapes du Projet :

#### 3.1 Exploration et Prétraitement

- Analyse exploratoire :
  - comprendre la structure du dataset (dimensions, types de variables).
  - identifier les classes cibles présentes et analyser leur distribution.
  - visualiser la longueur des textes, la fréquence des mots, les mots dominants par classe (nuages de mots).
- Nettoyage du texte :
  - gérer les doublons, traiter les valeurs manquantes et nettoyer les caractères spéciaux, liens, mentions, hashtags, émojis.
  - prétraiter les données textuelles : tokenisation, suppression des stopwords, lemmatisation, et stemming.
  - identifier les mots rares et fréquents pour effectuer un nettoyage supplémentaire en fonction de leur fréquence.
- Détection et gestion du déséquilibre des classes :
  - observer la répartition des classes et appliquer, si nécessaire, des techniques comme le sur/sous-échantillonnage ou la pondération.
- Création et transformation de variables utiles (Data Augmentation/Feature engineering), en adaptant les choix aux caractéristiques du dataset utilisé.

#### 3.2 Mise en œuvre de modèles supervisés (ML)

- Représentation du texte :
  - utiliser plusieurs méthodes pour transformer le texte en vecteurs numériques, y compris Bag of Words, TF-IDF, et Word Embeddings (Word2Vec, GloVe, etc.).
- Modèles à entraîner :
  - entraîner et comparer une Régression logistique ainsi que deux autres modèles de votre choix, parmi les algorithmes classiques de Machine Learning (SVM, Forêt Aléatoire, Naive Bayes, Gradient Boosting, etc.).
  - utiliser GridSearchCV et RandomizedSearchCV pour rechercher automatiquement les meilleurs hyperparamètres pour chaque modèle.
- Évaluation et Analyse des Erreurs :
  - utiliser des métriques de classification (précision, rappel, score F1, AUC-ROC, etc.) pour mesurer la performance et la capacité de généralisation de vos modèles.
  - analyser les erreurs de classification avec la matrice de confusion, et identifier les types de phrases les plus difficiles à classer.

#### 3.3 Approche Deep Learning (DL)

- Utilisation d'Embeddings :
  - utiliser des embeddings contextuels pré-entraînés (ex. BERT, RoBERTa) pour transformer les textes en vecteurs.
- Modèles Avancés de Réseaux de Neurones :
  - entraîner un modèle RNN simple, puis un modèle LSTM ou GRU, afin de comparer leurs performances et leur capacité à conserver le contexte dans les textes.

- fine-tuner un modèle de Transformer (ex. BERT) sur votre jeu de données afin d'évaluer l'apport de l'apprentissage par transfert.
- Hyperparamètres et Optimisation :
  - effectuer une recherche d'hyperparamètres pour optimiser vos modèles, en ajustant notamment le learning rate, la taille des batchs, le nombre d'époques, et le taux de dropout si nécessaire.
- Évaluation des modèles :
  - évaluer chaque modèle (RNN, LSTM/GRU, Transformer) à l'aide des métriques de classification utilisées dans la partie ML.
  - comparer ces performances à celles obtenues avec les modèles de Machine Learning classique.

### **3.4 Analyse Non Supervisée et Clustering**

- Clustering des textes :
  - utiliser des méthodes de clustering (K-means, DBSCAN) pour regrouper les textes en clusters.
  - Appliquer la réduction de dimensionnalité (PCA ou t-SNE) pour visualiser les clusters et interpréter leurs contenus.
- Analyse Thématique (Topic Modeling) :
  - utiliser LDA (Latent Dirichlet Allocation) pour découvrir les thèmes récurrents du corpus et analyser leur pertinence.

## PROJET 2 – Vision par Ordinateur (Computer Vision)

**Durée :** 6 à 7 semaines

**Travail :** en groupes de 5 étudiants

### Objectif général :

Ce projet vise à reproduire une problématique concrète de Vision par Ordinateur, en concevant un système capable de reconnaître automatiquement l'expression faciale d'un individu, parmi une liste d'émotions prédéfinies (joie, colère, tristesse, surprise, peur, neutralité, etc.).

Chaque groupe travaillera sur un jeu d'images dédié, et devra mettre en place une chaîne complète de traitement d'images, en explorant aussi bien des méthodes classiques de Machine Learning (ML) que des approches modernes de Deep Learning (DL).

Le projet inclut également :

- la prise en compte d'un éventuel déséquilibre entre les différentes classes d'expressions,
- l'optimisation des hyperparamètres pour améliorer les performances des modèles,
- ainsi qu'une analyse non supervisée permettant d'étudier les regroupements naturels présents dans les données visuelles.

### 1. Description générale du projet :

Vous réaliserez une chaîne complète de traitement d'images dédiée à la reconnaissance d'expressions faciales. Le projet est structuré en plusieurs étapes permettant d'explorer les principales phases d'un système de Vision par Ordinateur :

1. Sélection du jeu d'images (par groupe).
2. Prétraitement des images.
3. Mise en œuvre de modèles supervisés (ML) basés sur des descripteurs d'images classiques.
4. Approche Deep Learning (DL) à l'aide de CNN et de modèles pré-entraînés.
5. Analyse non supervisée et Clustering.
6. Analyse du déséquilibre des classes et optimisation des modèles.
7. Rédaction du rapport final et présentation.

### 2. Jeux d'images proposés :

Chaque groupe doit sélectionner une base d'images pour son projet. Vous pouvez choisir l'une des bases suggérées ci-dessous ou proposer une autre base équivalente, à condition qu'elle soit adaptée à la reconnaissance d'expressions faciales et suffisamment volumineuse.

Bases d'images proposés :

1. **Dataset 1 :** [FER2013](#)  
35000 images de visages annotées selon 7 expressions (joie, colère, neutralité, etc.).
2. **Dataset 2 :** [RAF-DB \(Real-world Affective Faces Database\)](#)  
Images de visages en conditions réelles, annotées selon 7 émotions.
3. **Dataset 3 :** [ExpW Dataset](#)  
90000 images de visages annotées dans 7 catégories d'émotions.

**Remarque :** ces jeux de données sont disponibles sur Kaggle, assurez-vous de respecter les licences et conditions d'utilisation de chaque jeu de données avant de les exploiter. Les groupes souhaitant utiliser un autre dataset librement choisi doivent le faire valider par l'enseignant avant de commencer le projet.

### 3. Étapes du Projet :

#### 3.1 Exploration et Prétraitement

- Analyse exploratoire :
  - analyser la répartition des images pour chaque expression et identifier un éventuel déséquilibre entre les classes.
  - visualiser plusieurs exemples de visages pour chaque émotion afin de comprendre les variations d'apparence.
  - examiner la résolution des images, leur qualité, ainsi que la présence de bruit visuel, d'occlusions (lunettes, cheveux, mains) ou de variations de pose et d'éclairage.
- Prétraitement des images :
  - détecter et recadrer automatiquement les visages (ex. Haar Cascades, MTCNN) afin d'éliminer les zones inutiles de l'image.
  - identifier les images problématiques ou très bruitées (visages non détectés, flous extrêmes, mauvaises annotations, arrière-plans dominants, images corrompues), puis décider de les supprimer ou les corriger.
  - standardiser les images : redimensionnement, normalisation des valeurs de pixels, conversion éventuelle en niveaux de gris ou en RGB selon les modèles.
  - appliquer des techniques d'augmentation de données (rotations modérées, zoom, flips horizontaux, variations d'éclairage, ajout de bruit, léger flou) afin d'améliorer la robustesse du modèle face aux variations réelles de pose, de lumière et d'occlusions.
- Gestion du déséquilibre des classes :
  - appliquer, si nécessaire, des techniques comme le sur/sous-échantillonnage, la pondération des classes ou l'augmentation ciblée afin d'améliorer l'équilibre entre les catégories émotionnelles.

#### 3.2 Mise en œuvre de modèles supervisés (ML)

- Extraction de caractéristiques (Feature Extraction) :
  - utiliser des descripteurs d'images classiques pour transformer chaque visage en vecteurs numériques (ex. HOG, LBP, SIFT, ou des vecteurs d'intensité réduits via PCA).
  - comparer l'impact de différentes représentations visuelles sur les performances des modèles.
- Modèles à entraîner :
  - en se basant sur les descripteurs extraits, entraîner et comparer une Régression logistique ainsi que deux autres modèles de votre choix, parmi les algorithmes classiques de Machine Learning (SVM, KNN, Forêt Aléatoire, etc.).
  - utiliser GridSearchCV et RandomizedSearchCV pour rechercher automatiquement les meilleurs hyperparamètres pour chaque modèle.
- Évaluation et Analyse des Erreurs :
  - évaluer les performances à l'aide de métriques de classification (accuracy, précision, rappel, score F1) et analyser la capacité de généralisation de chaque modèle.
  - interpréter les erreurs via la matrice de confusion, en identifiant notamment les expressions les plus confondues entre elles (ex. peur vs surprise, colère vs dégoût).

#### 3.3 Approche Deep Learning (DL)

- Réseau de neurones entièrement connectés (MLP) :
  - implémenter un premier modèle MLP en aplatissant les images en vecteurs 1D afin d'établir une baseline Deep Learning simple.
  - analyser ses limites (perte d'information spatiale, sensibilité à la rotation et au bruit) et comparer ses performances à celles des CNN.



- Réseaux convolutifs (CNN) :
  - concevoir un modèle CNN composé de couches de convolution, d'activation et de pooling, afin d'extraire automatiquement des motifs visuels pertinents (bords, textures, formes du visage).
  - entraîner ce modèle, puis expérimenter différentes profondeurs, tailles de filtres et techniques de régularisation (dropout, batch normalization) afin d'améliorer ses performances.
- Apprentissage par transfert (Transfer Learning) :
  - utiliser un modèle pré-entraîné (ex. VGG16, ResNet, EfficientNet) comme extracteur de caractéristiques afin de bénéficier de représentations visuelles déjà apprises sur des bases d'images larges et variées.
  - adapter ce modèle à la tâche de reconnaissance d'expressions faciales en ajoutant une ou plusieurs couches finales adaptées au nombre de classes émotionnelles.
  - entraîner ou fine-tuner tout ou partie du modèle pour évaluer l'apport du transfert d'apprentissage par rapport au CNN.
- Hyperparamètres et optimisation :
  - effectuer une recherche d'hyperparamètres pour optimiser vos modèles, en ajustant notamment le learning rate, la taille des batchs, le nombre d'époques, ainsi que les paramètres de régularisation (dropout, batch normalization) si nécessaire.
- Évaluation des modèles :
  - évaluer les différents modèles (MLP, CNN, modèles pré-entraînés) à l'aide des métriques de classification classiques : accuracy, précision, rappel, score F1, ainsi que la matrice de confusion.
  - comparer les performances des différentes approches Deep Learning (MLP vs CNN vs Transfer Learning) afin d'identifier les architectures les plus efficaces pour la reconnaissance d'expressions faciales.

### 3.4 Analyse Non Supervisée et Clustering

- Clustering des images :
  - appliquer des méthodes comme K-means ou DBSCAN sur des descripteurs classiques ou sur des représentations extraites par un CNN ou un modèle pré-entraîné, afin de regrouper automatiquement les images similaires.
- Visualisation :
  - utiliser PCA, t-SNE ou UMAP pour projeter les images dans un espace 2D et observer la formation éventuelle de groupes naturels.
- Analyse :
  - interpréter les clusters obtenus pour identifier les émotions proches visuellement, les catégories difficiles à séparer, et l'impact de la pose, de l'éclairage ou des occlusions sur les regroupements.