

Computer Vision pour la lecture automatique de factures

Projet INGE 2 | Majeure IA – Promo 2026  
ESME Paris

Thomas KUSNIEREK

Baptiste BEHR

Benjamin SZUREK

Encadrant : Wajd Meskini



Virtualisation et lecture de documents est un enjeu majeur pour de nombreuses entreprises.

**Problématique** : Concevoir un système d'extraction automatique du montant total d'une facture, tout en garantissant sa fiabilité et reproductibilité.

**Objectif** : détecter le montant total d'une facture avec une précision supérieure à 90%.

Utilisation d'un dataset de 53 images et test avec 3 types de modèles

1 OpenCV + Tesseract Open CV + PyTesseract (traitement d'image en temps réel et un wrapper Python traitement de texte imprimé)

**Pré-traitement de l'image : binarisation, floutage, contours (filtre Canny).**  
Aussi un filtre **Regex** : `r'\d{1,3}(?:[. ,]\d{2})'` pour détecter valeurs numériques au format décimal.



Métriques montrant une erreur importante.

Avec filtre, 27 images sur 53 n'ont pas leur contour qui est détecté, ce qui explique en partie le 0% de réussite:  
**Résultat décevant.**

Métrique	Avec filtre	Sans filtre
Pourcentage de réussite (%)	40.385	0.000
Erreur absolue moyenne : MAE (€)	72.10	53.84

2 Hugging Face (Plateforme open source permettant de partager des modèles – utilisation d'un modèle existant ayant servi à un projet similaire)

HUGGING FACE

**Double-test** sur le modèle de HuggingFace :

- Détection du prix avec un **pré-process** sur l'image (agrandissement de l'image, conversion en niveau de gris)
- Utilisation en plus d'un **fallback** utilisant PyTesseract si aucun montant n'est détecté.

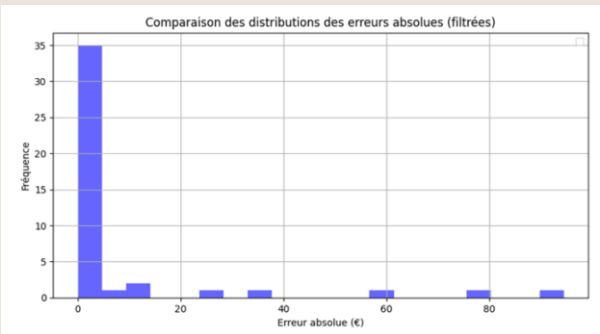


Résultat type

**L'accuracy** avec et sans fallback reste significativement la même.

Avec le fallback, on a quelques images qui n'arrivent pas à être traité parmi celle où le résultat n'était pas correct.

L'erreur est bien moins importante en considérant des valeurs tronquées, le modèle est **plutôt performant**

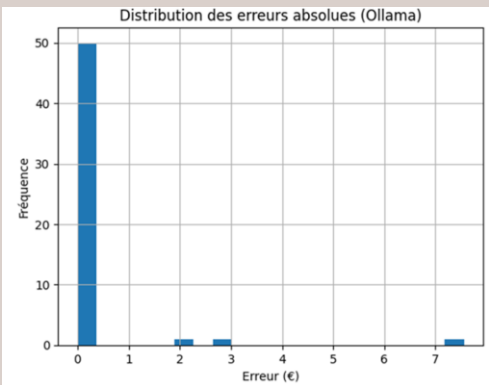


Métrique	Avec fallback	Sans fallback
Pourcentage de réussite (accuracy) (%)	59.62	62.00
MAE (€)	7.87	7.87

3 LLM Gemini – Serveur Ollama (LLM = type spécifique de modèle utilise du Deep Learning avancé pour générer du langage naturel).

Un LLM demande énormément de puissance de calcul. Le **CPU** d'un PC n'est pas conçu pour faire tourner un LLM. Cela est bien plus rapide avec un **GPU** (5 secondes contre 9 minutes).

On a envoyé un message à Ollama lui donnant le rôle d'analyser une facture et de renvoyer le montant total avec 2 décimales :



Métrique	Ollama
Pourcentage de réussite %	90.57
Erreur absolue moyenne : MAE (€)	0.24
Nombre d'erreurs de traitement sur 53 images :	0

Ce modèle est le **plus performant, résultat supérieur à 90 %**, ce qui est **attendu**.  
Cependant, méthode coûteuse en énergie, en mémoire, et non déterministique.

**Conclusion** : nos méthodes de détection du montant de notre facture deviennent plus performantes au fur et à mesure que le modèle devient complexe.