

REPUBLIQUE DU CAMEROUN
Paix-Travail-Patrie

MINISTRE DE L'ENSEIGNEMENT
SUPERIEUR

UNIVERSITE DE DSCHANG

FACULTE DES SCIENCES

DEPARTEMENT DE MATHÉMATIQUES
&
INFORMATIQUE



REPUBLIC OF CAMEROON
Peace-Work-Fatherland

MINISTRY OF HIGHER EDUCATION

UNIVERSITY OF DSCHANG

FACULTY OF SCIENCES

DEPARTMENT OF MATHEMATICS
&
COMPUTER SCIENCES

Détection de nids de poule sur le tronçon Dschang-Bafoussam

Par :

Nom et Prénom

DONGMO FEUDJIO Divin Virulence
KENMOGNE Ange Prisca

Matricule

CM-UDS-21SCI0878
CM-UDS-21SCI

Sous la direction de :

Nom et Prénom

Pr KENGNE TCHENDJI Vianney
YAWENI

Statut

Maître de Conférences
Chargé de TD

10 février 2026

- 1 Analyse du papier : You only look once : Unified, real-time object detection
- 2 Projet de détection de nids de poule sur le tronçon de route Dschang-Bafoussam

Analyse du papier : You Only Look Once (YOLO) I

Résumé et Contexte

Papier de Redmon et al. (2016) : Introduit YOLO pour détection en temps réel.

- **Approche** : Régression unique vs. pipelines complexes (R-CNN).
- **Points clés** :
 - Vitesse : 45 FPS (base), 155 FPS (Fast YOLO).
 - Précision : Double mAP temps réel, mais + erreurs localisation.

Analyse du papier : You Only Look Once (YOLO)

Système de détection YOLO

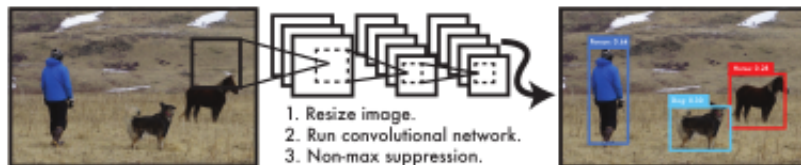


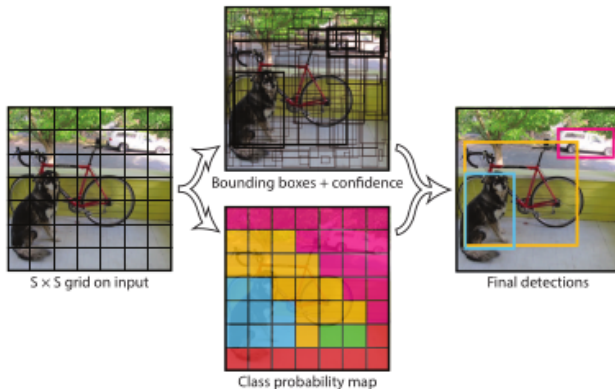
Figure – Système de détection de YOLO [1]

- Division de l'image en grille
- Prédiction simultanée par chaque cellule de chaque grille
- Post-traitement avec suppression non-maximale (NMS)

Analyse du papier : You Only Look Once (YOLO)

Division de l'image en grille (Grid Division)

L'image est divisée en grille $S \times S$ et si le centre d'un objet tombe dans une cellule de grille, cette cellule de grille est responsable de la détection de cet objet.



Analyse du papier : You Only Look Once (YOLO)

Prédiction simultanée par chaque cellule de chaque grille

- Chaque **cellule de grille** prédit B boîtes englobantes, des scores de confiance pour ces boîtes et C probabilités de classes conditionnelles.
- **Score** : $Pr(Class_i|Object) \times Pr(Object) \times IOU$.
- Chaque boîte englobante consiste en 5 prédictions : x, y, w, h, et la confiance
 - x,y : les coordonnées du centre de la boîte
 - w,h : la largeur et la hauteur de la boîte
 - la confiance : IOU entre la boîte prédite et toute boîte vérité terrain

Analyse du papier : You Only Look Once (YOLO)

Non-Maximum Suppression (NMS)

Après prédiction, plusieurs boîtes peuvent détecter le même objet.

Étapes du NMS :

- Trier les boîtes par ordre décroissant de **score de confiance**.
- Sélectionner la boîte avec le score le plus élevé.
- Supprimer toutes les boîtes avec **IoU** > **seuil** (généralement 0.5) avec celle-ci.
- Répéter jusqu'à épuisement.

Résultat : une seule détection finale par objet, sans doublons.

Analyse du papier : You Only Look Once (YOLO) I

Architecture du réseau

Ce modèle est implémenté comme un réseau neuronal convolutif.

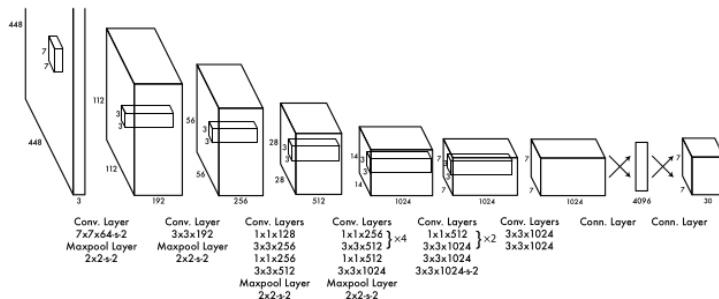


Figure – Modèle de détection de YOLO [1]

Analyse du papier : You Only Look Once (YOLO) II

Architecture du réseau

- **Entraînement :**

- Pré-entraînement des couches convolutives.
- Conversion du modèle pour la détection en ajoutant 4 couches convolutives et 2 couches entièrement connectées.
- Augmentation de la résolution d'entrée de 224x224 à 448x448 pour des informations visuelles fines.
- Utilisation d'une fonction de perte somme des carrés pondérée ($\lambda_{coord} = 5$, $\lambda_{noobj} = 0.5$).
- Prédiction de la racine carrée de w et h pour mieux gérer les erreurs sur petites boîtes.
- Assignment d'un prédicteur responsable par objet basé sur l'IOU le plus élevé.

- **Avantages :**

- Une seule évaluation du réseau pour prédire 98 boîtes et classes par image.

Analyse du papier : You Only Look Once (YOLO) III

Architecture du réseau

- Utilisation de NMS pour éliminer les détections multiples.
- Pas de pipeline complexe, rapide pour le streaming vidéo.
- **Limites :**
 - Contraintes spatiales : seulement 2 boîtes et 1 classe par cellule, limite les objets proches.
 - Difficulté avec petits objets groupés (ex. : troupeaux d'oiseaux).

Analyse du papier : You Only Look Once (YOLO) I

Fonction de perte

YOLO optimise une **fonction de perte multi-parties** basée sur l'erreur quadratique (sum-squared error). Cette fonction est composée de cinq termes distincts, chacun correspondant à un objectif d'apprentissage différent.

La fonction complète est la suivante :

Analyse du papier : You Only Look Once (YOLO) II

Fonction de perte

$$\begin{aligned} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

Analyse du papier : You Only Look Once (YOLO) I

Fonction de perte

Explication détaillée de chaque terme

- Erreur de localisation sur les coordonnées du centre (x , y) = premier terme
 - Calcule l'erreur quadratique (MSE) entre le centre prédit (\hat{x}_i, \hat{y}_i) et le centre réel (x_i, y_i).
 - x et y sont relatifs à la cellule de la grille (normalisés entre 0 et 1).
- Erreur de localisation sur la taille de la boîte (w , h) (deuxième terme)
 - Même principe, mais pour la largeur (w) et la hauteur (h).
 - On applique la **racine carrée** avant le calcul de l'erreur.

Analyse du papier : You Only Look Once (YOLO) II

Fonction de perte

- **Raison de la racine carrée** : une erreur de 10 pixels sur une petite boîte est beaucoup plus grave que sur une grande boîte. La racine carrée rend l'erreur proportionnellement plus sévère pour les petites tailles.
- **Erreur de confiance pour les boîtes contenant un objet (troisième terme)**
 - Erreur quadratique sur la **confiance** prédite (\hat{C}_i) par rapport à la vraie confiance ($C_i = \text{IOU}_{\text{pred}}^{\text{truth}}$).
 - Objectif : apprendre que, quand il y a un objet, la confiance doit être élevée (proche de 1 ou de l'IOU réel).
- **Erreur de confiance pour les boîtes sans objet (quatrième terme)**
 - Même erreur de confiance, mais pour les cellules/boîtes **sans objet** ($C_i = 0$).

Analyse du papier : You Only Look Once (YOLO) III

Fonction de perte

- Objectif : pénaliser les faux positifs (prédire un objet là où il n'y en a pas).
- **Erreur de classification (probabilités de classes)**
(cinquième terme)
 - Erreur quadratique sur les probabilités conditionnelles $\Pr(\text{Class}_c \mid \text{Objet})$.
 - On compare la probabilité prédite ($\hat{p}_i(c)$) à la vraie (1 pour la classe correcte, 0 sinon).
 - Utilise MSE pour simplicité (les versions modernes passent à la cross-entropy).

Analyse du papier : You Only Look Once (YOLO) I

Fonction de perte

Résumé des objectifs de la fonction de perte

- **Localisation** (termes 1 et 2) → apprendre à bien placer et dimensionner les boîtes (fortement pondérée).
- **Confiance avec objet** (terme 3) → apprendre à être confiant quand il y a un objet.
- **Confiance sans objet** (terme 4) → apprendre à ne pas être confiant quand il n'y a rien (pénalité réduite).
- **Classification** (terme 5) → apprendre à identifier correctement la classe de l'objet (uniquement quand objet présent).

Cette conception astucieuse est la raison pour laquelle YOLO parvient à équilibrer vitesse et précision malgré sa simplicité.

Analyse du papier : You Only Look Once (YOLO)

En conclusion

En conclusion :

- YOLO est un modèle unifié, simple à construire, entraîné directement sur images complètes.
- Optimisé pour performances de détection ; rapide, précis et généralisable.
- Pousse l'état de l'art en détection temps réel ; idéal pour applications nécessitant robustesse.

- 1 Analyse du papier : You only look once : Unified, real-time object detection
- 2 Projet de détection de nids de poule sur le tronçon de route Dschang-Bafoussam

Objectifs du projet

Contexte et problématique

Problématique principale

Les nids-de-poule constituent un danger majeur sur les routes camerounaises, notamment sur l'axe **Dschang–Bafoussam**.

Objectifs du projet :

- Détecter automatiquement les nids-de-poule en temps réel à partir de vidéos de dashcam
- Avertir le conducteur (alerte visuelle / sonore)
- Cartographier les zones à risque pour les autorités
- Quantifier les dommages routiers

Choix du dataset public

- Dataset : Potholes-Detection-YOLOv8
- Auteur : Angga Dwi Sunarto
- Plateforme : Kaggle
- Lien : <https://www.kaggle.com/datasets/anggadwisunarto/potholes-detection-yolov8>
- Format natif : YOLOv8
- Nombre d'images :
 - Train : 1581
 - Valid : 396

Import et traitement via Roboflow

- Création d'un compte sur [Roboflow](#)
- Projet : Pothole Cameroon
- Upload du ZIP Kaggle complet
- Redimensionnement : 512×512
- Nouvelle version du dataset : images_ok
- Split final : **70% – 15% – 15%**
- Export au format **YOLOv8**

Lien du dataset final :

[https://app.roboflow.com/pothole-detection-wipyl/
pothole-detection-dschang-bafous/1](https://app.roboflow.com/pothole-detection-wipyl/pothole-detection-dschang-bafous/1)

Environnement d'entraînement

Configuration utilisée

- Plateforme : Google Colab (GPU)
- GPU : Tesla T4
- Bibliothèque : Ultralytics YOLOv8
- Modèle de base : **yolov8n.pt** (nano – très léger)
- Nombre d'epochs : 100

Notebook complet : [https:](https://github.com/VirlenceDongmo/Pothole_Detection/blob/master/Code/pothole_notebook/Pothole_detection.ipynb)

[//github.com/VirlenceDongmo/Pothole_Detection/blob/master/Code/pothole_notebook/Pothole_detection.ipynb](https://github.com/VirlenceDongmo/Pothole_Detection/blob/master/Code/pothole_notebook/Pothole_detection.ipynb)

Interprétation des métriques clés

- $mAP@0.5 = 75.8\%$ → excellent
- Precision = 76.8%
- Recall = 68.3%
- $mAP@0.5 : 0.95 = 48.8\%$

Interprétation

- Très bon compromis précision / rapidité
- Peu de faux positifs → alertes fiables
- Recall perfectible sur petits / mal éclairés potholes

Application web de démonstration

- Framework : **Flask**
- Fonctionnalités :
 - Upload vidéo
 - Détection en temps réel
 - Visualisation des résultats
- Modèle chargé : best.pt

Application web de démonstration



Figure – Page d'accueil de l'application

Synthèse et résultats obtenus

Points forts

- mAP@0.5 75.8% sur un dataset mixte
- Déploiement fonctionnel (Flask)
- Code et données publics sur GitHub

Lien du projet complet :  [VirulenceDongmo / Pothole_Detection](#)

Perspectives d'amélioration

- Collecte et annotation d'images locales
- Augmentation des données (pluie, nuit, poussière, ombre)
- Intégration en temps réel dans un véhicule
- Développement d'une application mobile
- Création d'une carte interactive des nids-de-poule (base de données géolocalisée)

Références

- [1] Joseph Redmon et al. "You only look once : Unified, real-time object detection". In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 779-788.