

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

**Rapport de projet**

Par Lahcene Zinnour

DÉPARTEMENT DE MATHÉMATIQUES

ET

D'INFORMATIQUE

© Zinnour, Automne 2019

# Table des matières

[Table des matières ii](#_Toc29606500)

[Liste des tableaux iii](#_Toc29606501)

[Liste des figures iv](#_Toc29606502)

[No table of figures entries found. iv](#_Toc29606503)

[Introduction v](#_Toc29606504)

[Méthode proposée v](#_Toc29606505)

[PWCNET vi](#_Toc29606506)

[Apprentissage par transfert viii](#_Toc29606507)

[Résultats expérimentaux ix](#_Toc29606508)

[Conclusion x](#_Toc29606509)

[Bibliographie xi](#_Toc29606510)

[xii](#_Toc29606511)

[xiii](#_Toc29606512)

# Liste des tableaux

**No table of figures entries found.**

# Liste des figures

[Figure 1. L'architecture proposée. vi](#_Toc29606513)

[Figure 2. Champ vectoriel et sa carte de couleurs correspondante vi](#_Toc29606514)

[Figure 3. Architecture réseau de PWC-Net. vii](#_Toc29606515)

[Figure 4. Flux optique ix](#_Toc29606516)

[Figure 5. Entrainement de réseau ix](#_Toc29606517)

# Introduction

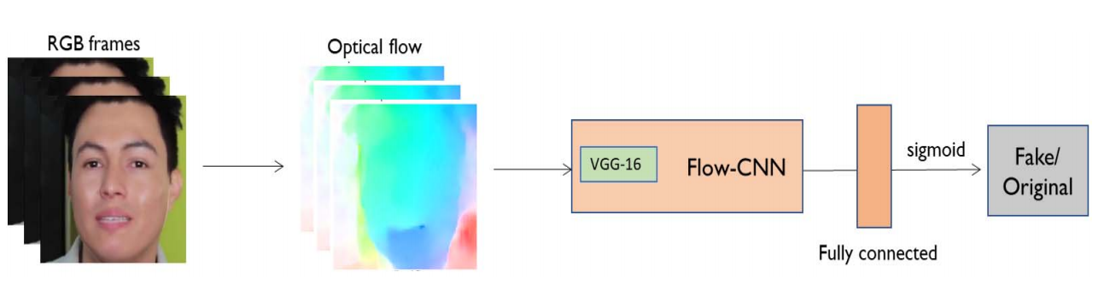
Deepfake (issu de «l'apprentissage profond» et «faux») est une technique qui peut superposer des images de visage d'une personne cible à une vidéo d'une personne source pour créer une vidéo de la personne cible faisant ou disant des choses que la personne source fait. Les modèles d'apprentissage profond tels que les auto-encodeurs et les réseaux adverses génératifs ont été largement appliqués dans le domaine de la vision par ordinateur pour résoudre divers problèmes. Ces modèles ont également été utilisés par des algorithmes de deepfake pour examiner les expressions faciales et les mouvements d'une personne et synthétiser des images faciales d'une autre personne faisant des expressions et des mouvements analogues. Les algorithmes Deepfake nécessitent normalement une grande quantité de données d'images et de vidéos pour entraîner les modèles à créer des images et des vidéos photo-réalistes. Étant donné que des personnalités publiques telles que des célébrités et des politiciens peuvent avoir un grand nombre de vidéos et d'images disponibles en ligne, elles sont les premières cibles des deepfakes. Il existe également une utilisation positive des deepfakes tels que la création de voix de ceux qui ont perdu leur voix ou la mise à jour d'épisodes de films. Cependant, le nombre d'utilisations malveillantes de deepfakes domine largement celui des utilisations positives.

La méthode proposée est l'adoption de champs de flux optiques pour exploiter les éventuelles dissemblances inter-trames. Un tel indice est ensuite utilisé comme caractéristique à apprendre par les classificateurs CNN.

# Méthode proposée

Le flux optique est un champ vectoriel qui est calculé sur deux images consécutives f (t) et f (t + 1) pour extraire le mouvement apparent entre l'observateur et la scène elle-même. En particulier, notre hypothèse est que le flux optique est capable d'exploiter les écarts de mouvement entre les images créées synthétiquement par rapport à celles générées naturellement par une caméra vidéo. Elle devrait être plus appréciable dans les matrices de flux optique, l'introduction de mouvements faux et inhabituels des lèvres, des yeux et en général du visage entier. Ainsi, pour cette raison, pour chaque image f (t), à un certain instant t, un flux direct OF (f (t), f (t + 1)) est extrait à l'aide du modèle CNN de flux optique appelé PWC-Net.

Cette technique est basée sur trois techniques traitées par le CNN lui-même pour estimer le flux optique. Successivement (figure 1), le flux avant calculé OF (f (t), f (t + 1)) est donné en entrée à un CNN semi-entrainé nommé Flow-CNN, basé sur un réseau pré-entrainé (l’architecture de réseau VGG16).

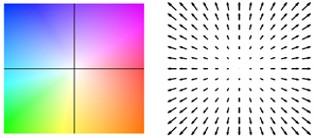


1. L'architecture proposée.

# PWCNET

Le but de l'estimation du flux optique est de générer une carte 2D dense à valeur réelle (vecteur u, v) du mouvement se produisant à partir de deux images vidéo. Ces informations peuvent être très utiles lorsque nous essayons de résoudre des problèmes de vision par ordinateur tels que le suivi d'objets, la reconnaissance d'actions, la segmentation d'objets vidéo, etc.

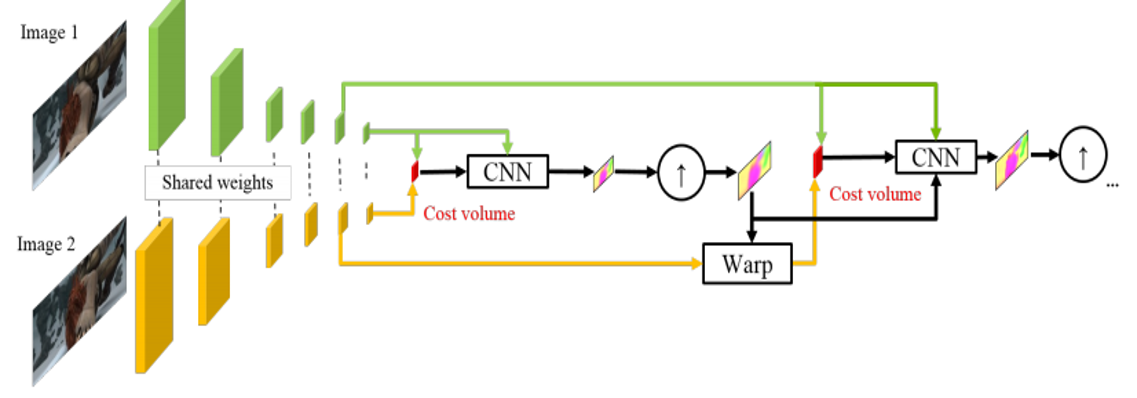
La figure (b) indique le codage couleur utilisé pour la visualisation des champ de vecteurs (u, v). Habituellement, l'orientation vectorielle est représentée par la teinte de couleurtandis que la longueur du vecteur est codée par la saturation des couleurs:



1. Champ vectoriel et sa carte de couleurs correspondante

Selon [2018a], PWC Net améliore FlowNet2 [2016a] en ajoutant des connaissances de domaine à la conception du réseau. L'idée de base derrière l'estimation du flux optique est qu'un pixel conservera la majeure partie de sa luminosité au fil du temps malgré un changement de position d'une image à la suivante (constance de "luminosité"). Nous pouvons saisir un petit patch autour d'un pixel dans l'image vidéo 1 et trouver un autre petit patch dans l'image vidéo 2 qui maximisera certaines fonctions des deux patchs. Glisser ce patch sur l'ensemble de l'image 1, à la recherche d'un pic, génère ce qu'on appelle un volume des coûts (le C PWC). Cette technique est assez robuste (invariante au changement de couleur) mais coûte cher à calculer. Dans certains cas, nous pouvons avoir besoin d'un correctif assez important pour réduire le nombre de faux positifs dans l'image 1, augmentant encore la complexité.

Pour alléger le coût de génération du volume de coût, la première optimisation consiste à utiliser le traitement pyramidal (le P dans PWC). L'utilisation d'une image de résolution inférieure permet d'effectuer la recherche en faisant glisser un patch plus petit de l'image 1 sur une version plus petite de l'image 2, ce qui donne un vecteur de mouvement plus petit, puis utilise ces informations comme indice pour effectuer une recherche plus ciblée au niveau de résolution suivant dans la pyramide. Cette estimation de mouvement multi-échelle peut être réalisée dans l'image ou dans les caractéristiques (en utilisant la fonction downscaled feature maps générées par un réseau de neurones convolutifs). En pratique, PWC warps (le W dans PWC) l'image 1 en utilisant une version suréchantillonnée du flux de mouvement estimé à une résolution inférieure car cela conduira à rechercher un incrément de mouvement plus petit au niveau de résolution supérieur suivant de la pyramide.



1. Architecture réseau de PWC-Net.

# Apprentissage par transfert

Un modèle pré- entraîné est un réseau enregistré qui a été précédemment formé sur un grand ensemble de données, généralement sur une tâche de classification d'images à grande échelle. Nous utilisons le modèle pré-entraîné comme il est, ou le transfert d'utiliser l'apprentissage pour personnaliser ce modèle à une tâche donnée.

L'intuition derrière l'apprentissage par transfert est que si un modèle formé sur un ensemble de données suffisamment large et général, ce modèle servira effectivement de modèle générique du monde visuel. Nous pouvons ensuite profiter de ces caractéristiques d'entités apprises sans avoir à recommencer à zéro la formation d'un grand modèle sur un grand ensemble de données.

Dans ce travail, nous avons utilisé la méthode Fine-Tuning pour l'apprentissage par transfert. Le Fine-Tuning généralement, consiste à garder les poids de toutes les couches des réseaux de neurones pré-entrainé (sur le jeu de données A) à l'exception de l'avant-dernière couche et à entrainer le réseau de neurones sur le jeu de données B, juste pour apprendre les représentations sur l'avant-dernier couche. Nous remplaçons généralement la dernière couche (softmax) par une autre de notre choix (selon le nombre de sorties dont nous avons besoin pour le nouveau problème, donc dans notre cas, nous avons des données pour 2 classe différente, nous devons remplacer les dernières couches softmax par une couche sigmoid à 1 seul nœud).

Voici les étapes:

(1) prendre le VGG16 déjà entrainer sur ImageNet,

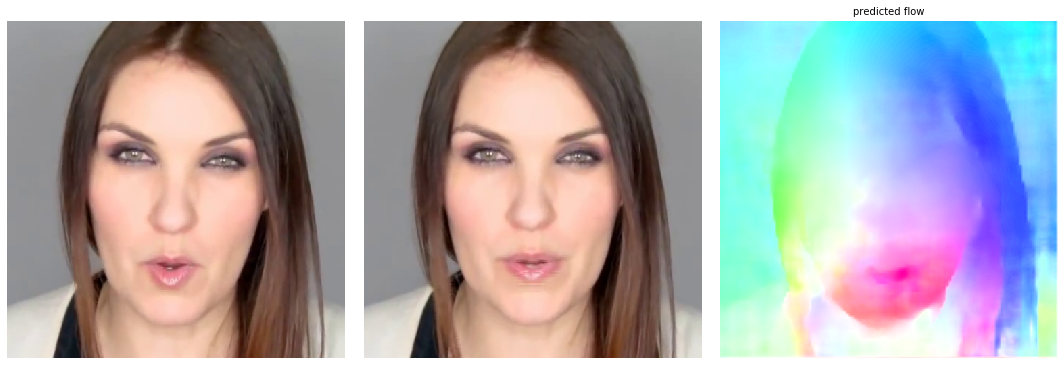
(2) fixer toutes les couches sauf les trois dernières couches entièrement connectées

(3) remplacer les trois dernières couches et

(4) entraîner les derniers et le sigmoid sur l'ensemble de données de flux optique pour en savoir plus sur les caractéristiques de la vidéo.

# Résultats expérimentaux

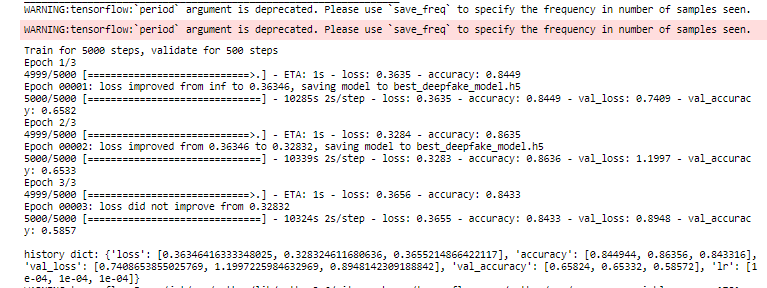
Dans cette section, des résultats expérimentaux sont présentés pour évaluer la qualité du concept proposé. Sur la figure 4, un exemple des champs de flux optique calculés sur deux images consécutives pour une vidéo originale.



1. Flux optique

Le réseau VGG16 a été entrainer sur des images de taille 120 × 120 pixels contenant le visage; pour l’entrainement, j’ai utilisé l'optimiseur Adam avec un taux d'apprentissage de 10−4, et batch égale à 256. J'ai utilisé l'ensemble de données FaceForensics ++; l'ensemble se compose de 450 séquences vidéo originales qui ont été manipulées avec une méthode de manipulation de visage automatisée Face2Face. 360 des vidéos sont utilisées pour l’entrainement, 45 pour la validation et 45 autres pour les tests.

La sortie de la phase de l’entrainement Figure 5



1. Entrainement de réseau

Exemple de prédire la classe d’un vidéo :





# Conclusion

Dans ce travail, l'idée d'exploiter les différences de champ de flux optique comme un indice pour discriminer entre les vidéos deepfake et les vidéos originales a été introduite et étudiée. Il s'agit d'une tentative très innovante de prendre en compte d'éventuelles anomalies dans la dimension temporelle de la séquence. Dans ces expériences, pour résoudre le problème de l'utilisation d'un réseau pré-entraine, les vecteurs de mouvement ont été représentés comme une image à 3 canaux, puis considérés comme une entrée pour un réseau neuronal. les résultats, obtenus sur le jeu de données FaceForensics ++ avec différents types de réseaux, sont très prometteurs et montrent que ce type de fonctionnalité semble pouvoir détecter la vidéo deepfake.

# Bibliographie

Deep Learning for Deepfakes Creation and Detection [Thanh Thi Nguyen et al]

PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume[Deqing Sun et al]

Deepfake Video Detection through Optical Flow based CNN [Irene Amerini et al]

https://github.com/philferriere/tfoptflow

# 

# 