Fully Convolutional Networks for Semantic Segmentation

Réalisé par :
ANDRIAMAHERISOA Liantsoa

Plan

- Étude de l’article : *(Fully Convolutional Networks for Semantic Segmentation)*
- Travaux Réalisés :

  - Prise en main du dataset
  - Prétraitement
  - Création du réseau de neurone
  - Entrainement
  - Performance

- Difficultés

Étude de l’article

Article : *fully convolutional Networks for Semantic Segmentation*

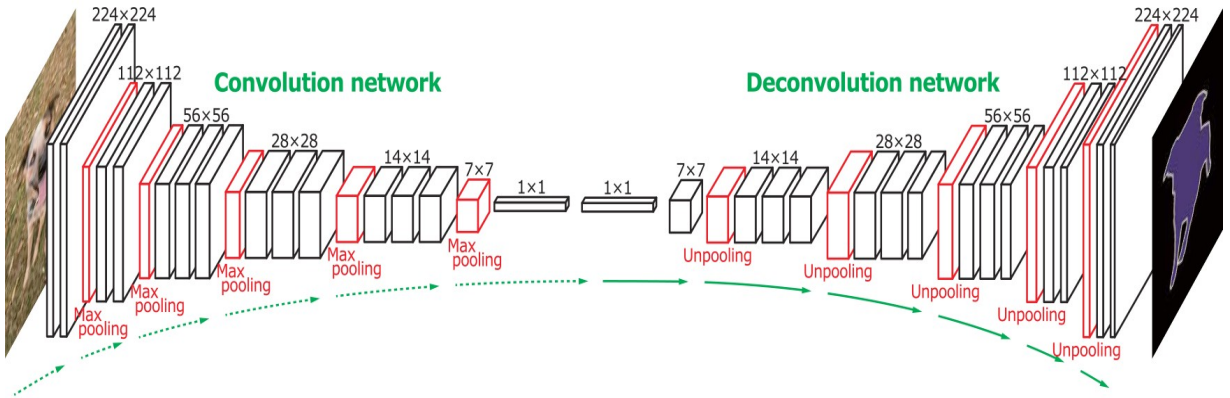
Auteurs : Jonathan Long, Evan Shelhamer, Trevor Darrell

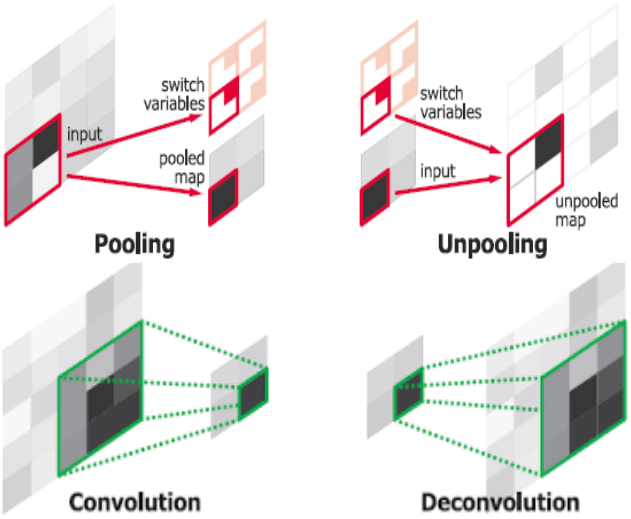
Année de publication : 2016

Problématique : Construire à partir d’une architecture CNN, des réseaux « entièrement convolutifs »

Étude de l’article

Fully convolutional networks

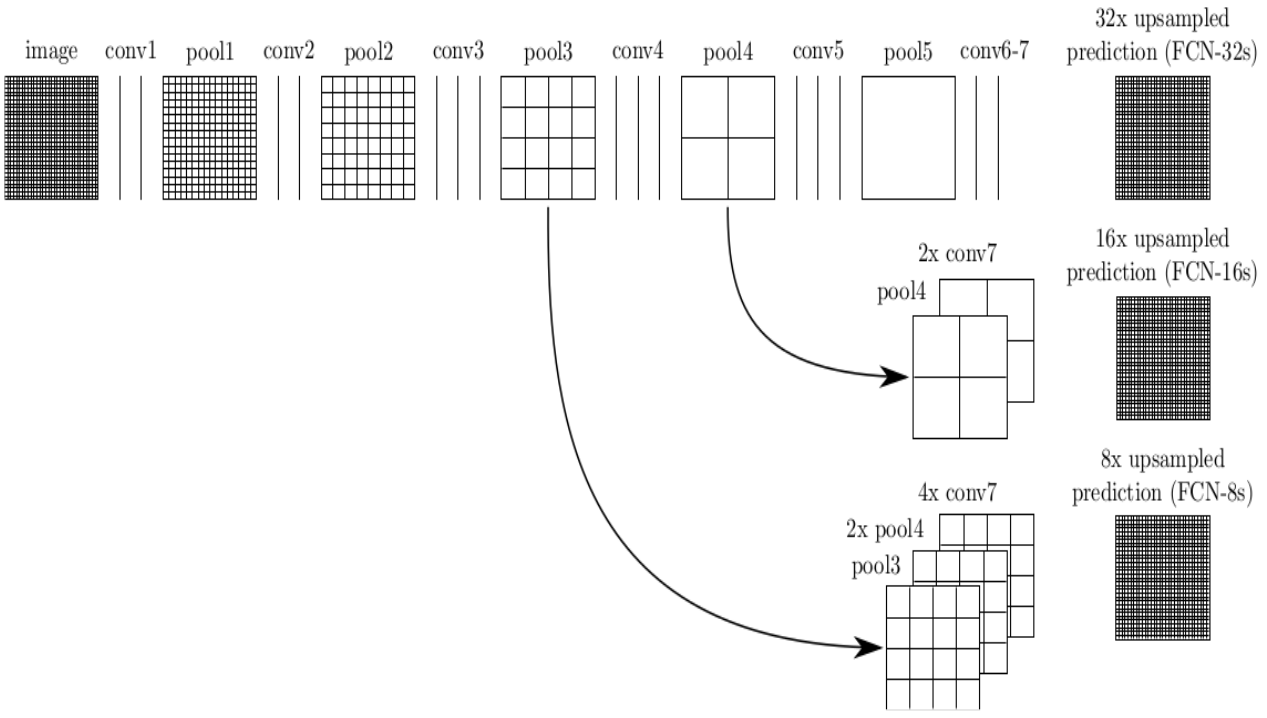


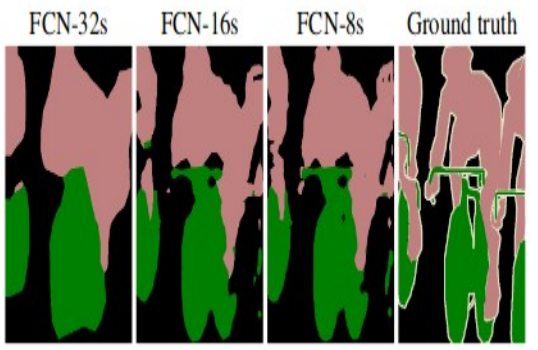


Transfer Learning

VggNet , AlexNet, GoogLeNet

Étude de l’article





Étude de l’article

Comparaison des performances entre FCN-32s, FCN-16s, FCN-8s

Datasets : PASCAL VOC

	pixel acc.	mean acc.
FCN-32s-fixed	83.0	59.7
FCN-32s	89.1	73.3
FCN-16s	90.0	75.7
FCN-8s	**90.3**	**75.9**

Étude de l'article

Ils ont testé leurs architectures FCN sur différents datasets et voici leurs résultats

SIFT FLOW

	pixel acc.	mean acc.	mean IU	f.w. IU	geom. acc.
Liu *et al.* [25]	76.7	-	-	-	-
Tighe *et al.* [36]	-	-	-	-	90.8
Tighe *et al.* [37] 1	75.6	41.1	-	-	-
Tighe *et al.* [37] 2	78.6	39.2	-	-	-
Farabet *et al.* [9] 1	72.3	50.8	-	-	-
Farabet *et al.* [9] 2	78.5	29.6	-	-	-
Pinheiro *et al.* [31]	77.7	29.8	-	-	-
FCN-16s	**85.2**	**51.7**	39.5	76.1	**94.3**

PASCAL VOC

	mean IU VOC2011 test	mean IU VOC2012 test	inference time
R-CNN [12]	47.9	-	-
SDS [17]	52.6	51.6	~ 50 s
FCN-8s	**62.7**	**62.2**	~ **175 ms**

PASCAL CONTEXT

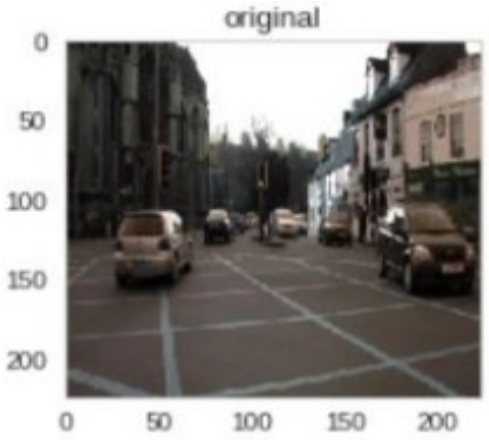
59 class	pixel acc.	mean acc.	mean IU	f.w. IU
O_2P	-	-	18.1	-
CFM	-	-	31.5	-
FCN-32s	65.4	47.2	35.1	50.3
FCN-16s	66.8	49.6	37.6	52.3
FCN-8s	**67.0**	**50.7**	**37.8**	**52.5**

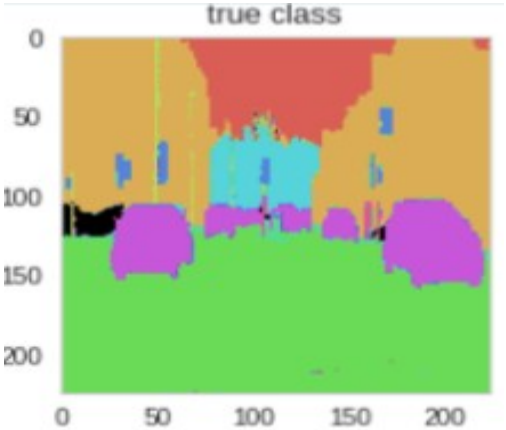
Travaux Réalisés

- Prise en main du dataset

  - 367 images de scènes extérieurs

  - la taille des images et d’environ 360 x 480 pixels





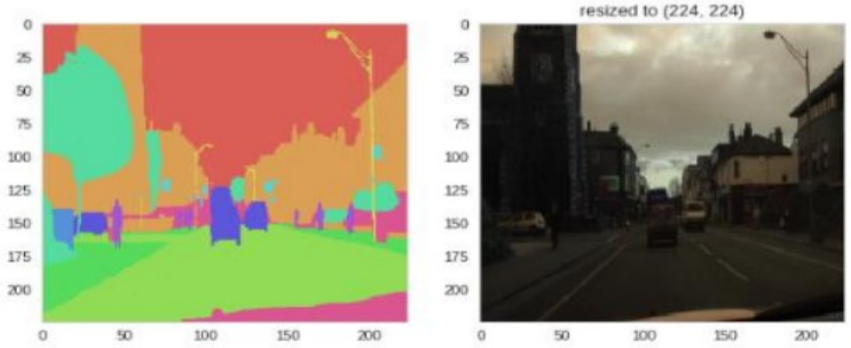
12 classes

1 classe = 1couleur

Travaux Réalisés

- Prétraitement

  - Redimensionnement et égalisation de la taille de toutes les images (360, 480, 3) → (224, 224, 3)



  - Partitionnement du dataset :

    311 images d’entrainement

    56 images de validation

Travaux Réalisés

- Création du réseau de neurone

  - Le modèle créé est d'architecture FCN, empile des couches de covonlution avec des couches de dé-convolution, pour arriver à réaliser la tâche de la segmentation, en faisant appel au poids du réseau **VGG16**.

Partie 1 : Convolution

5 Bloques

Tous les filtres utilisées dans les couches de convolutions sont de taille 3 x 3, avec un nombre allant de 64 sur le premier bloque, jusqu'à 512.

Une couche de Maxpooling appliquée sur les images avec un filtre de taille 2 x 2

```
img_input = Input(shape=(input_height,input_width, 3)) ## Assume 224,224,3

## Block 1
x = Conv2D(64, (3, 3), activation='relu', padding='same', name='block1_conv1', data_format=IMAGE_ORDERING )(img_input)
x = Conv2D(64, (3, 3), activation='relu', padding='same', name='block1_conv2', data_format=IMAGE_ORDERING )(x)
x = MaxPooling2D((2, 2), strides=(2, 2), name='block1_pool', data_format=IMAGE_ORDERING )(x)
f1 = x

# Block 2
x = Conv2D(128, (3, 3), activation='relu', padding='same', name='block2_conv1', data_format=IMAGE_ORDERING )(x)
x = Conv2D(128, (3, 3), activation='relu', padding='same', name='block2_conv2', data_format=IMAGE_ORDERING )(x)
x = MaxPooling2D((2, 2), strides=(2, 2), name='block2_pool', data_format=IMAGE_ORDERING )(x)
f2 = x

# Block 3
```

Travaux Réalisés

Partie 2 : Déconvolution

- Agrandir les images, et avoir une sortie de même taille que l'entrée

- Couche de déconvolution utilisant l'architecture du réseau VGG16, et faisant appel au poids de ce réseau pré-entrainé

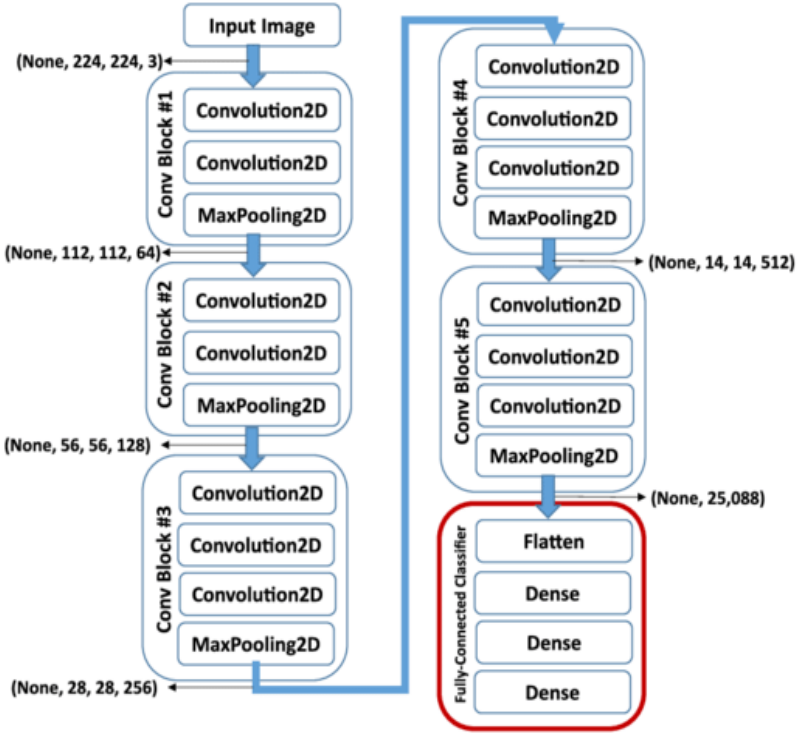
```
vgg  = Model(  img_input , x  )
vgg.load_weights(VGG_Weights_path)
```

```
x = ( Conv2DTranspose( nClasses , kernel_size=(2,2) ,  strides=(2,2) , use_bias=False)(x)
```

Travaux Réalisés

Architecture VGG16

Utiliser par l'équipe VGG pendant la compétition ILSVRC en 2014



Le modèle VGG-16 à 16 couches de convolutions et il est entrainé sur la base de données ImageNet.

5 couche de pooling

Fonction d'activation : Relu

Travaux Réalisés

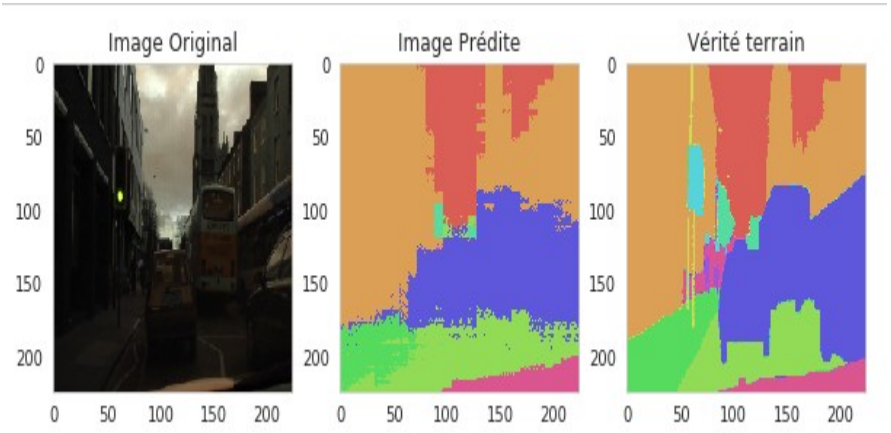
- Entrainement

```
Train on 170 samples, validate on 30 samples
Epoch 1/10
170/170 [==============================] - 1028s 6s/step - loss: 1.9977 - accuracy: 0.3105 - val_loss: 1.4790 - va
l_accuracy: 0.5622
Epoch 2/10
170/170 [==============================] - 717s 4s/step - loss: 1.1261 - accuracy: 0.6290 - val_loss: 0.9451 - val
_accuracy: 0.7064
Epoch 3/10
170/170 [==============================] - 674s 4s/step - loss: 0.8657 - accuracy: 0.7327 - val_loss: 0.8556 - val
_accuracy: 0.7330
Epoch 4/10
170/170 [==============================] - 669s 4s/step - loss: 0.7707 - accuracy: 0.7655 - val_loss: 0.8382 - val
_accuracy: 0.7475
Epoch 5/10
170/170 [==============================] - 663s 4s/step - loss: 0.7025 - accuracy: 0.7871 - val_loss: 0.7929 - val
_accuracy: 0.7508
Epoch 6/10
170/170 [==============================] - 684s 4s/step - loss: 0.6587 - accuracy: 0.8046 - val_loss: 0.7339 - val
_accuracy: 0.7760
Epoch 7/10
 94/170 [===============>..............] - ETA: 5:53 - loss: 0.6436 - accuracy: 0.8099 ETA: 5:48 - loss: 0.6412 -
accuracy:
```

Travaux Réalisés

- Performance

Données de validation pour faire une segmentation



IOU = Intersection Over Union

```
class 00: #TP= 32601, #FP=  2385, #FN= 3349, IoU=0.850
class 01: #TP= 90385, #FP= 18779, #FN= 5992, IoU=0.785
class 02: #TP=     0, #FP=     3, #FN= 2914, IoU=0.000
class 03: #TP= 26423, #FP=  8965, #FN= 2997, IoU=0.688
class 04: #TP= 11053, #FP=  1001, #FN= 9256, IoU=0.519
class 05: #TP=  1693, #FP=   384, #FN= 3256, IoU=0.317
class 06: #TP=     0, #FP=     0, #FN= 4300, IoU=0.000
class 07: #TP=     0, #FP=     1, #FN=  595, IoU=0.000
class 08: #TP= 32338, #FP= 16240, #FN= 6333, IoU=0.589
class 09: #TP=     0, #FP=     0, #FN= 2690, IoU=0.000
class 10: #TP=     0, #FP=     0, #FN=  572, IoU=0.000
class 11: #TP=  7816, #FP=   813, #FN= 6317, IoU=0.523

_________________
Moyenne IoU: 0.356
```

Les Difficultés

- Travailler seul

- Les calculs

MERCI !!!