

Activité1_Classifier une image avec GoogLeNet

Cet exemple montre comment classer une image à l'aide des réseaux de neurones convolutifs profonds pré-entraîné GoogLeNet, Alexnet ou Mobilenetv2.

GoogLeNet a été entraîné sur plus d'un million d'images et peut classer les images en 1000 catégories d'objets (tels que le clavier, la tasse à café, le crayon et de nombreux animaux). Le réseau a appris des représentations riches en caractéristiques pour un large éventail d'images. Le réseau prend une image en entrée, puis génère une étiquette pour l'objet dans l'image avec les probabilités pour chacune des catégories d'objets.

Télécharger le réseau pré-entraîné

Téléchargez le réseau pré-entraîné. Attention, cette étape nécessite le package de prise en charge du modèle Deep Learning Toolbox™ pour le réseau GoogLeNet ou celui d'Alexnet ou autre. Si vous n'avez pas installé les packages de support requis, le logiciel fournit un lien de téléchargement.

Vous pouvez également choisir de charger un autre réseau pré-entraîné pour la classification des images. Pour essayer un autre réseau pré-entraîné, ouvrez cet exemple dans MATLAB® et sélectionnez-en un autre. Par exemple, vous pouvez essayer squeezenet, un réseau qui est encore plus rapide que GoogLeNet. Vous pouvez exécuter cet exemple avec d'autres réseaux pré-entraînés. Pour obtenir la liste de tous les réseaux disponibles, reportez-vous à la section [Load Pretrained Networks](#).

Code : nnet = mobilenetv2;

L'image que vous souhaitez classer doit avoir la même taille que la taille d'entrée du réseau. Pour GoogLeNet, le premier élément de la propriété Layers du réseau est la couche d'entrée d'image. La taille d'entrée réseau est la propriété InputSize de la couche d'entrée d'image.

Code :

```
inputSize = nnet.Layers(1).InputSize
```

Affichage sur le LiveEditor

```
inputSize = 1x3
```

```
224 224 3
```

Le dernier élément de la propriété Layers (Couches) est la couche de classification en sortie. La propriété ClassNames de cette couche contient les noms des classes apprises par le réseau. Voir 10 noms de classe aléatoires sur un total de 1000.

Code :

```
classNames = nnet.Layers(end).ClassNames;
```

```
numClasses = numel(classNames);
```

```
disp(classNames(randperm(numClasses,10)))
```

Affichage sur le LiveEditor

```
{'speedboat'      }  
{'window screen' }  
{'isopod'        }  
{'wooden spoon'  }  
{'lipstick'      }  
{'drake'         }  
{'hyena'         }  
{'dumbbell'      }  
{'strawberry'    }  
{'custard apple' }
```

Lire et redimensionner l'image

Lisez et affichez l'image que vous souhaitez classer.

Code :

```
l = imread('http://www.monveloestunique.fr/wp-content/uploads/2018/12/649-vtt-electrique-wayscral-sporty755.jpg');
```

```
figure
```

```
imshow(l)
```

Affichage sur le LiveEditor



Affichez la taille de l'image. L'image mesure 3522 x 5819 pixels et dispose de trois canaux de couleur (RVB).

Code :

```
size(l)
```

Affichage sur le LiveEditor

```
ans = 1x3
```

```
3522    5819     3
```

Redimensionnez l'image à la taille d'entrée du réseau à l'aide de la fonction `imresize`. Ce redimensionnement modifie légèrement les proportions de l'image.

Code :

```
I = imresize(I,inputSize(1:2));  
figure  
imshow(I)
```

Affichage sur le LiveEditor



En fonction de vos besoins d'application, vous souhaitez peut-être redimensionner l'image d'une manière différente. Par exemple, vous pouvez recadrer le coin supérieur gauche de l'image à l'aide de `I(1 :inputSize(1),1 :inputSize(2),:)`. Si vous disposez de la boîte™ à outils de traitement d'image, vous pouvez utiliser la fonction `imcrop`.

Classifier l'image

Classez l'image et calculez les probabilités de classe à l'aide de la fonction "classify". Le réseau classe correctement l'image comme un « Mountain Bike ». Un réseau de classification est entraîné à générer une étiquette unique pour chaque image d'entrée, même lorsque l'image contient plusieurs objets.

Code :

```
[label,scores] = classify(nnet,I);  
Label
```

Affichage sur le LiveEditor

```
label = categorical  
    mountain bike
```

Affichez l'image avec l'étiquette prédite et la probabilité prédite que l'image ait cette étiquette.

Code :

```
figure  
imshow(I)  
title(string(label) + ", " + num2str(100*scores(classNames == label),3) + "%");
```

Affichage sur le LiveEditor

mountain bike, 54.6%



Afficher les meilleures prédictions

Affichez les cinq principales étiquettes prédites et leurs probabilités associées sous forme d'histogramme. Étant donné que le réseau classe les images dans de nombreuses catégories d'objets et que de nombreuses catégories sont similaires, il est courant de prendre en compte les cinq principales précisions lors de l'évaluation des réseaux. Le réseau classe l'image comme un « Mountain Bike » avec une forte probabilité 54,6%.

Code :

```
[~,idx] = sort(scores,'descend');  
idx = idx(5:-1:1);  
classNamesTop = nnet.Layers(end).ClassNames(idx);  
scoresTop = scores(idx);
```

figure

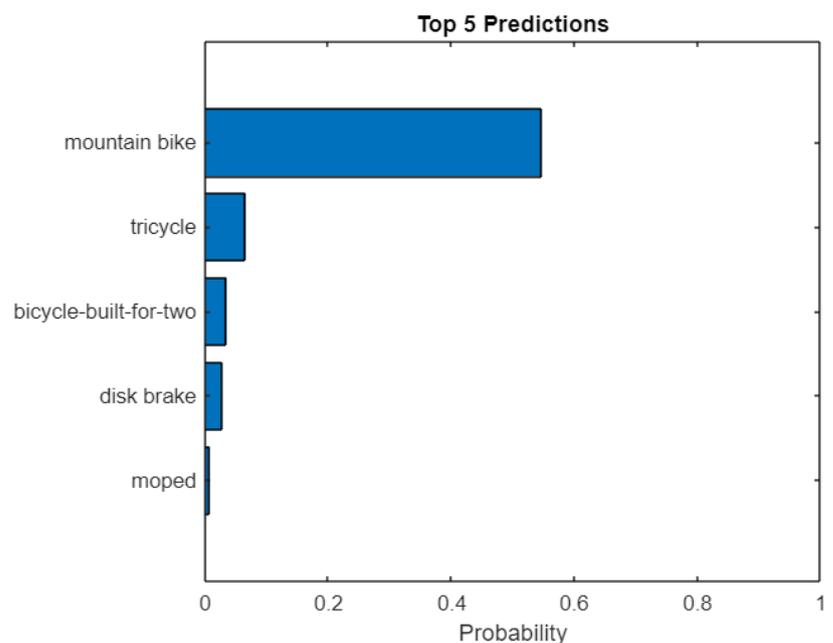
```
barh(scoresTop)
```

```
xlim([0 1])
```

```
title('Top 5 Predictions')
```

```
xlabel('Probability')
```

```
yticklabels(classNamesTop)
```



Affichage sur le LiveEditor

Questions :

Modifier le réseau en remplaçant « mobilenetv2 » par googlenet puis alexnet dans la première ligne de code.

1. Que constatez-vous sur la qualité des prédictions ?
2. Quel réseau semble le mieux entraîné ?
3. En recherchant les caractéristiques des 3 réseaux, d'après vous pourquoi ?

Modifiez l'image du vélo par une autre image en remplaçant son lien par celui que vous voulez, par exemple celui d'une souris ou d'un chien...

4. Que constatez-vous ?
5. Quel réseau semble le plus performant ?
6. Quels biais d'après vous existe dans ces différents réseaux ?