

# Architecture Convolutionnelle 2

Bouyrou Charles, Houitte Difanrsé Rebecca

# Réseau de neurones convolutifs (CNN)

- Définition
- Couches de convolution
- Couches de pooling

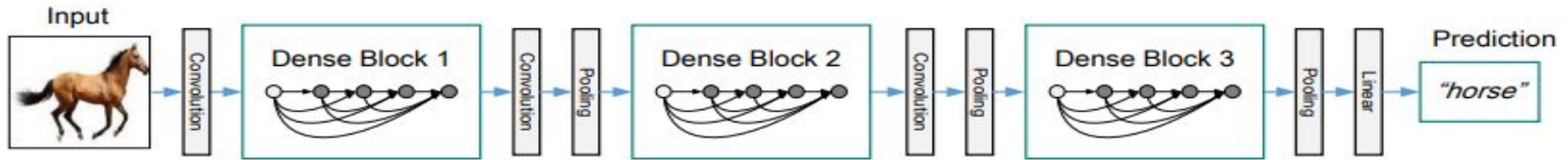
# Exemples de CNN

- Densely Connected Convolutional Networks (DenseNet)
- mobileNET
- Neural architecture search (NAS) et efficient Net
- SNN

# DenseNet

# Présentation générale du DenseNet

- CNN présenté en 2017, composé de blocs connectés
- Utilisé pour la classification d'images, la détection d'objets, la segmentation sémantique



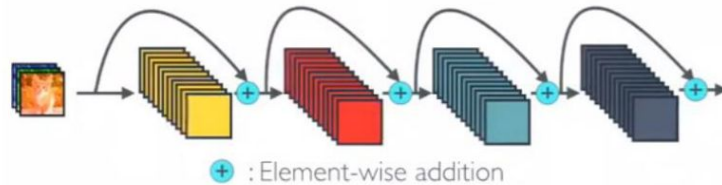
Exemple de DenseNet composé de 3 blocs denses, les couches entre les blocs sont appelés couches de transition et servent à modifier la taille des feature maps

# Bloc Dense

- Couches de convolutions 3x3 : extraction des caractéristiques, suivies d'une activation ReLU (modèle non linéaire)
- Batch normalisation
- Concaténations des caractéristiques : les sorties de toutes les couches précédentes sont concaténées puis transmises à la couche suivante
- Dropout

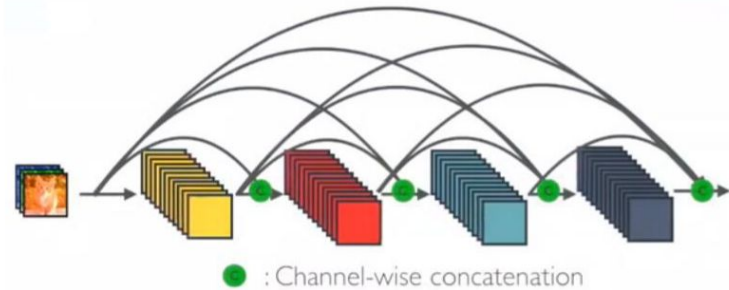
# Chanel wise concatenation (concaténation par canal)

**ResNet**



$$x^l = H^l(x^{l-1}) + x^{l-1}$$

**DenseNet**



$$x^l = H^l([x^0, x^1, \dots, x^{l-1}])$$

# Channel wise concatenation (concaténation par canal)

Avantage :

- Meilleure propagation du gradient
- Utilisation efficace des caractéristiques

Inconvénient :

- Les feature maps doivent avoir la même taille → bloc + couches de transition
- Complexité du modèle



# Réduction de la complexité du modèle

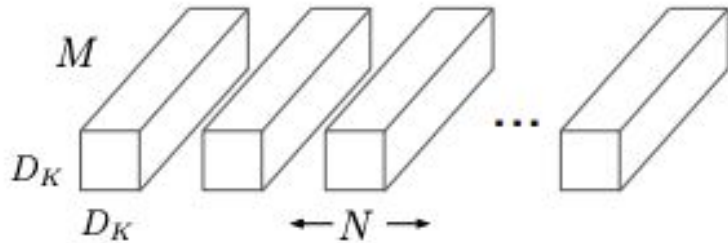
- $k$  (taux de croissance du réseau) = nb de feature maps produit par couche  
entrée de la couche  $\ell+1 = k_0 + k*\ell$
- Bottleneck layer (couche d'étranglement) : Conv 1x1 utilisé avant les conv 3x3
- Couches de transition : conv 1x1 et pooling pour rendre la modèle plus compact

# MobileNet

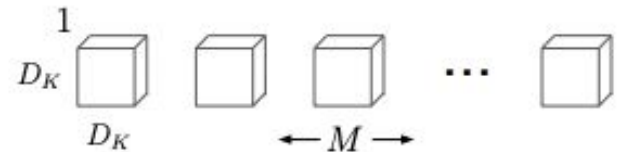
# MobileNET : la genèse

- Période cruciale dans le développement de modèle légers, efficaces et rapides
- CNN léger, efficace avec une grande précision
- Grand besoin dans des domaines comme la robotique, les voitures autonomes
- 2 options :
  - Compresser des réseaux pré entraînés
  - Entraîner des petits réseaux from scratch

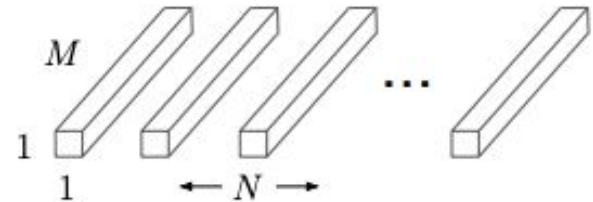
# Architecture CNN classique vs MobileNET



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c)  $1 \times 1$  Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

# Coûts computationnels CNN classique vs MobileNET

CNN Classique

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F$$

VS

MobileNET

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F$$

Gain

$$\frac{1}{N} + \frac{1}{D_K^2}$$

N : nombre de canaux en sortie

D<sub>k</sub> : taille du filtre

# Variantes de MobileNET

Introduction de 2  
hyperparamètres :

$\alpha$  et  $\rho$

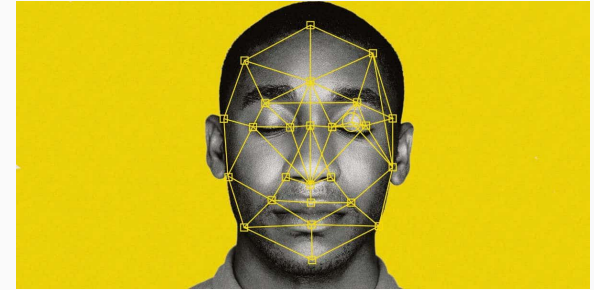
- $\alpha$  : width multiplier. Rôle : réduire la taille des inputs et outputs. Entre 0 et 1.
- $\rho$  : resolution multiplier. Rôle : moduler la resolution des inputs et outputs. Entre 0 et 1



Coût

$$D_K \cdot D_K \cdot \alpha M \cdot \rho D_F \cdot \rho D_F + \alpha M \cdot \alpha N \cdot \rho D_F \cdot \rho D_F$$

Gain d'environ :  $\alpha^2$  et  $\rho^2$

# Tâches réalisées par les réseaux MobileNET



<b>Input image</b>		
<b>Generic classification</b>	<b>Dog</b>	<b>Dog</b>
<b>Fine-grained classification</b>	<b>greyhound</b>	<b>Whippet</b>

# NAS et EfficientNet



# Neural architecture search ( Recherche d'Architectures Neuronales)

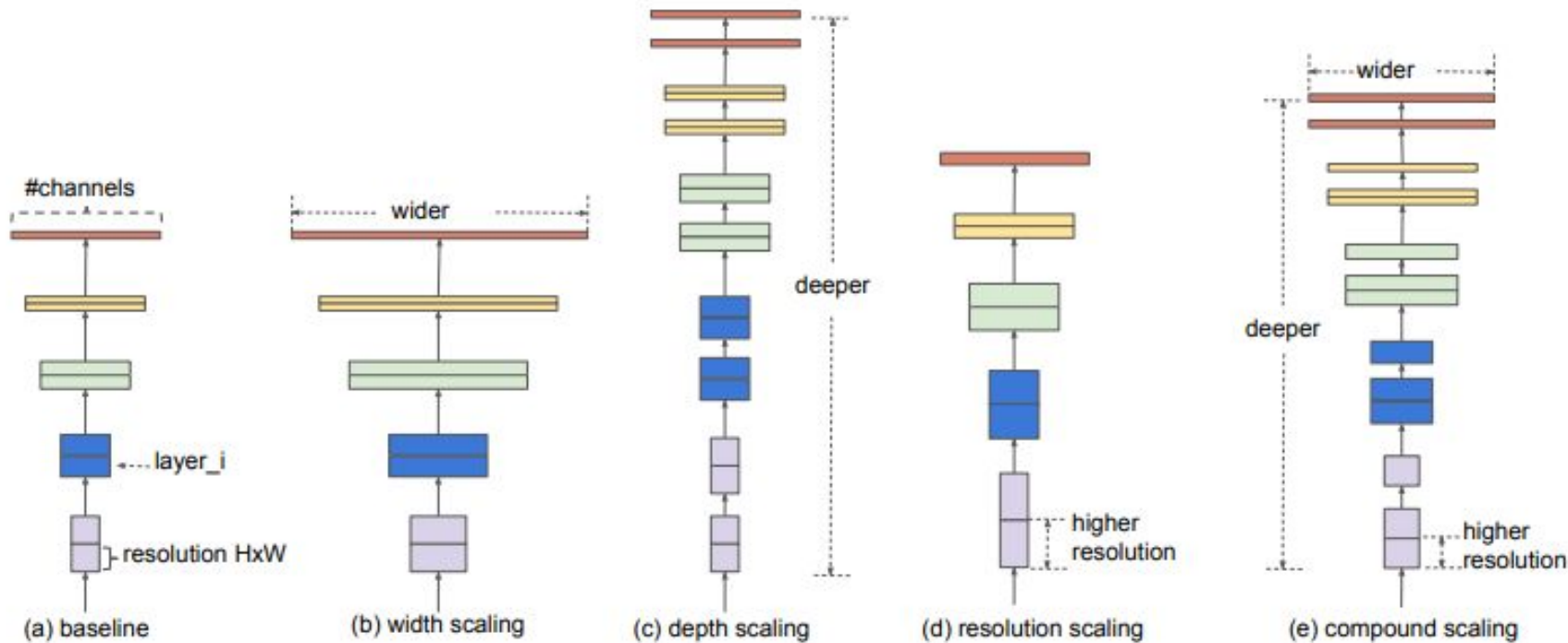
Vise à automatiser la conception des architectures de réseaux de neurones

- l'espace de recherche : ensemble des architectures candidates et opérations qui les composent
- la stratégie de recherche : apprentissage par renforcement, la descente de gradient
- le critère (ou méthode) d'évaluation : précision de la classification, temps d'exécution, la consommation d'énergie

# EfficientNet

Présenté en 2019 par Google AI, approche de mise à l'échelle appelée "compound scaling" : optimisation de la taille du réseau en ajustant simultanément la largeur, la profondeur et la résolution

- Augmentation de la taille : mieux capturer des motifs et des caractéristiques plus complexes (amélioration de la précision),
- Réduction de la taille : modèle plus léger, adapté aux environnements à ressources limitées, avec éventuellement de la perte de précision



# Fonctionnement

- Espace de recherche : modèle de base , coefficient composé phi fixé en fonction de la complexité du modèle souhaité
- Stratégie de recherche : grid search pour trouver les 3 dim optimales,
- Critère d'évaluation : selon la tâche spécifique

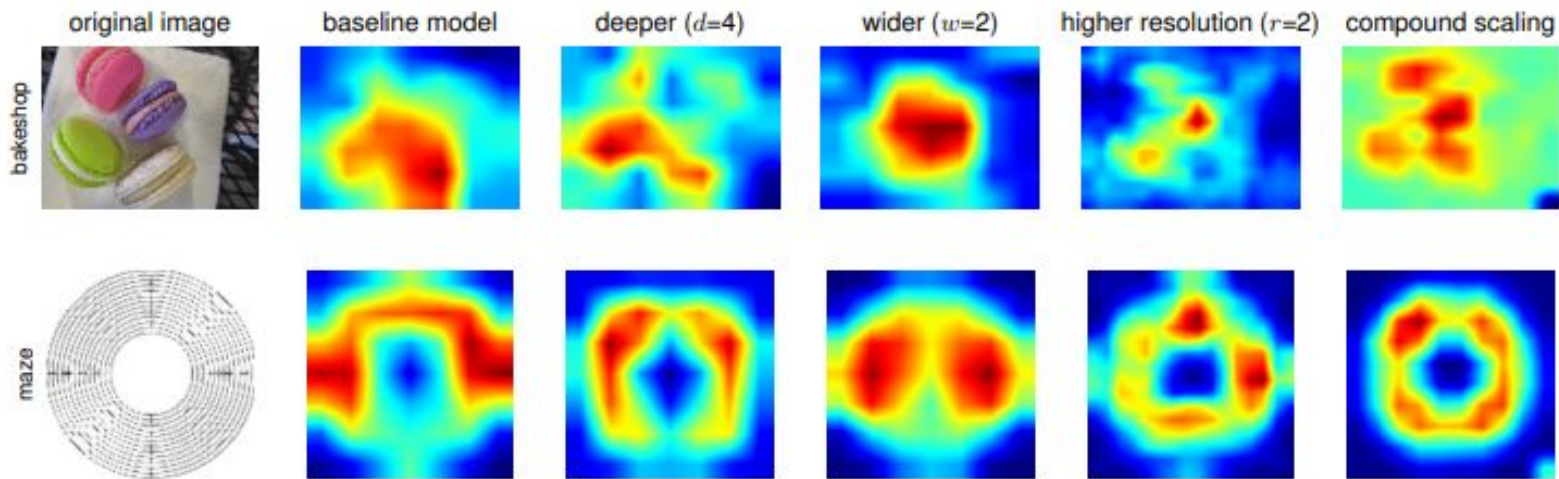
$$\text{depth: } d = \alpha^\phi$$

$$\text{width: } w = \beta^\phi$$

$$\text{resolution: } r = \gamma^\phi$$

$$\text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$$

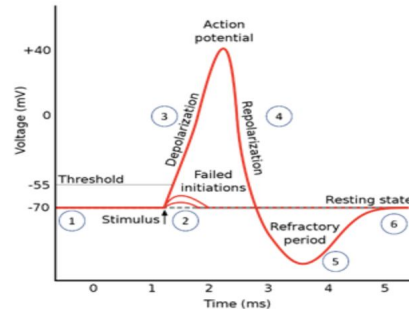
$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$



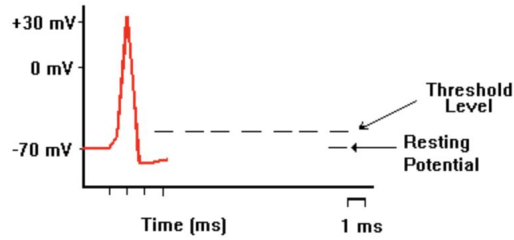
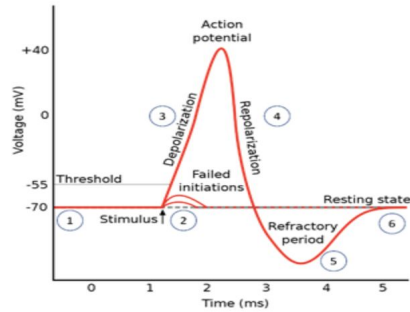
# SNN

# SNN : Spike Neural Networks

- Evolution des réseaux de neurones artificiels
- La transmission d'informations entre neurones repose sur le comportement biologique naturel du neurone



# SNN : Spike Neural Networks



Représentation du potentiel d'action

- 1) Repos
- 2) Stimuli faible
- 3) Dépolarisation
- 4) Repolarisation
- 5) Période réfractaire



# ANN vs SNN

- Fonctions d'activation
- Apprentissage par descente de gradient + rétropropagation (ex)
- Énergivore quand profond
- Efficace pour des tâches nécessitant une grande précisions et des gros calculs comme le NLP et la computer vision.

- Fonctions d'activation
- Apprentissage basée sur des règles dépendantes du temps (ex : ordre de déclenchements de potentiels d'action)
- Nécessite du calcul que lors de la réception de spike
- Prometteur pour les applications de capteur en temps réel, robotique neuromorphique et systèmes embarqués