

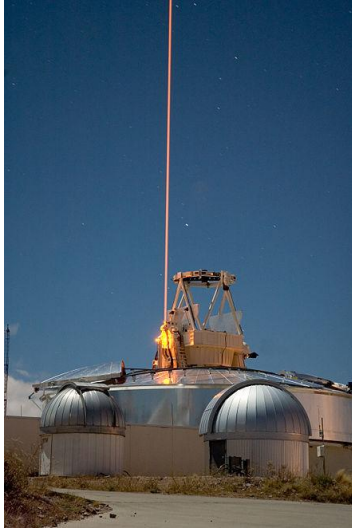


# Classification de formes en 3D : VoxNet

D'après le papier de Daniel Maturana et  
Sebastian Scherer : *VoxNet: A 3D Convolutional Neural  
Network for Real-Time Object Recognition*

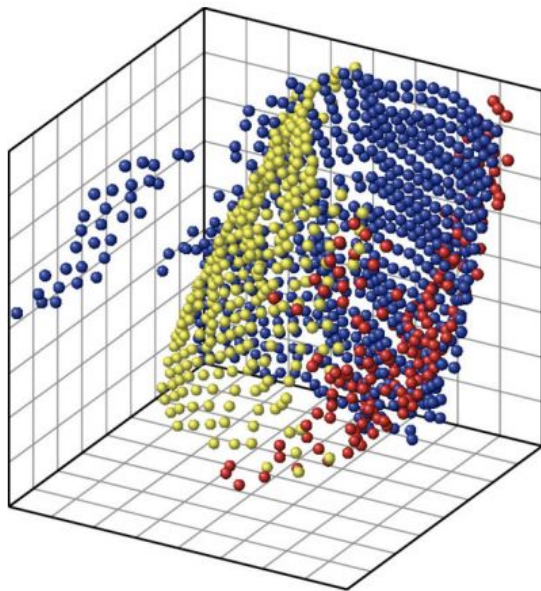
# L'architecture VoxNet : description du projet

- Problématique générale

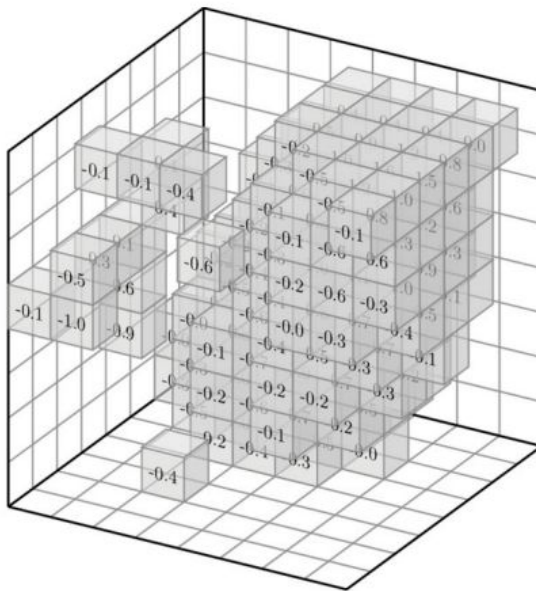


# Comment représenter une forme 3D?

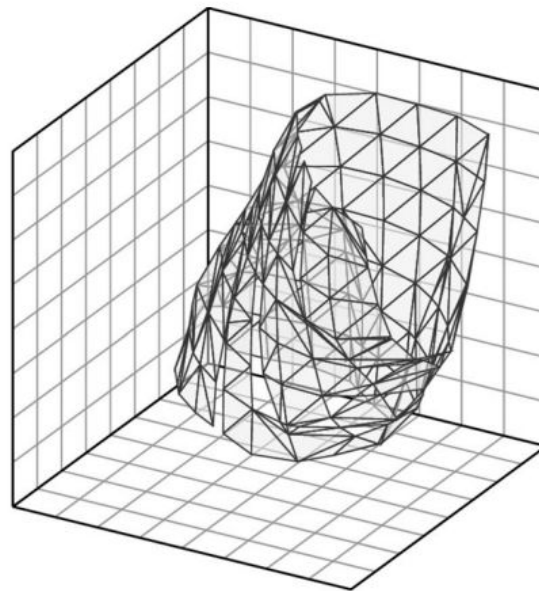
Nuage de points 3D



Voxels



Maille polygonale

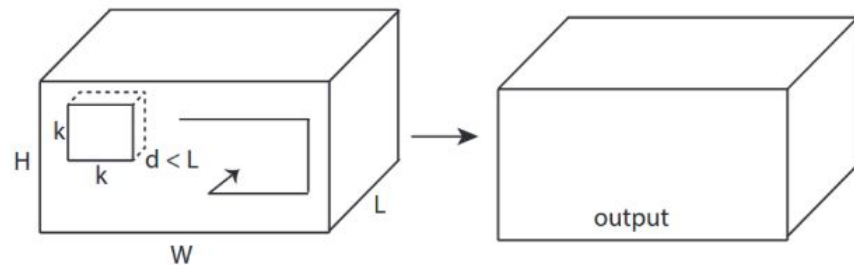
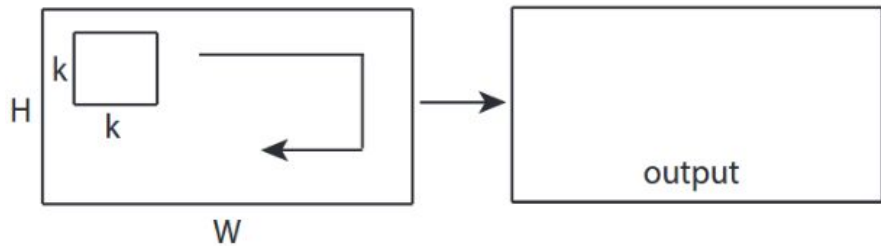


# Grilles d'occupation



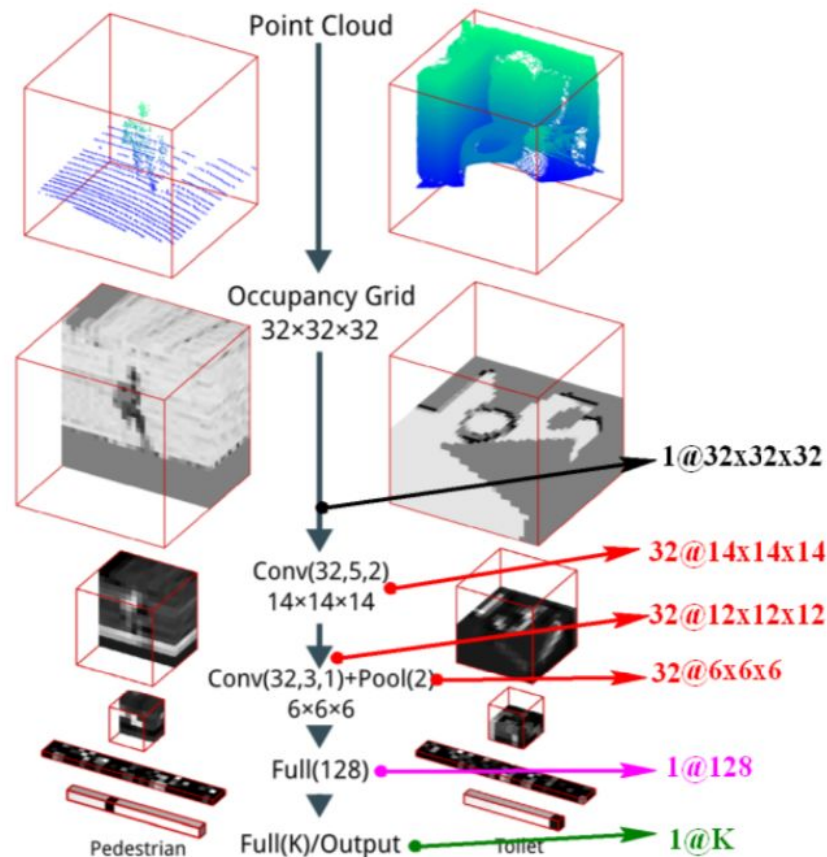
# Implémentation

- 3D-CNN



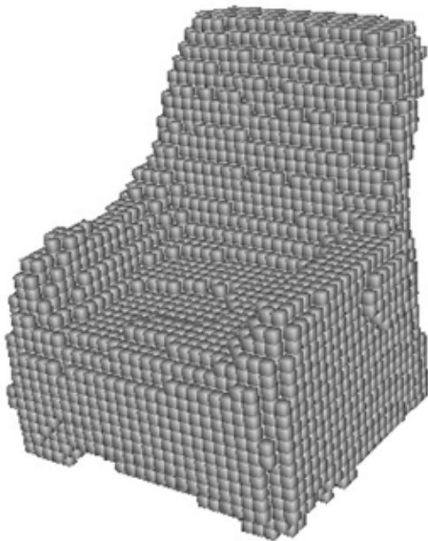
# Implémentation

- Les différentes couches



# Implémentation

Le Dataset finalement utilisé: ModelNet



# Résultats



	ModelNet10	ModelNet20	ModelNet30	ModelNet40
SGD	0.92	/	/	0.83
Adam	0.96	0.94	0.88	0.83*

- Effet réel du choix de l'optimiseur
- Précision cohérente à mesure que le nombre de classes augmente



# Résultats



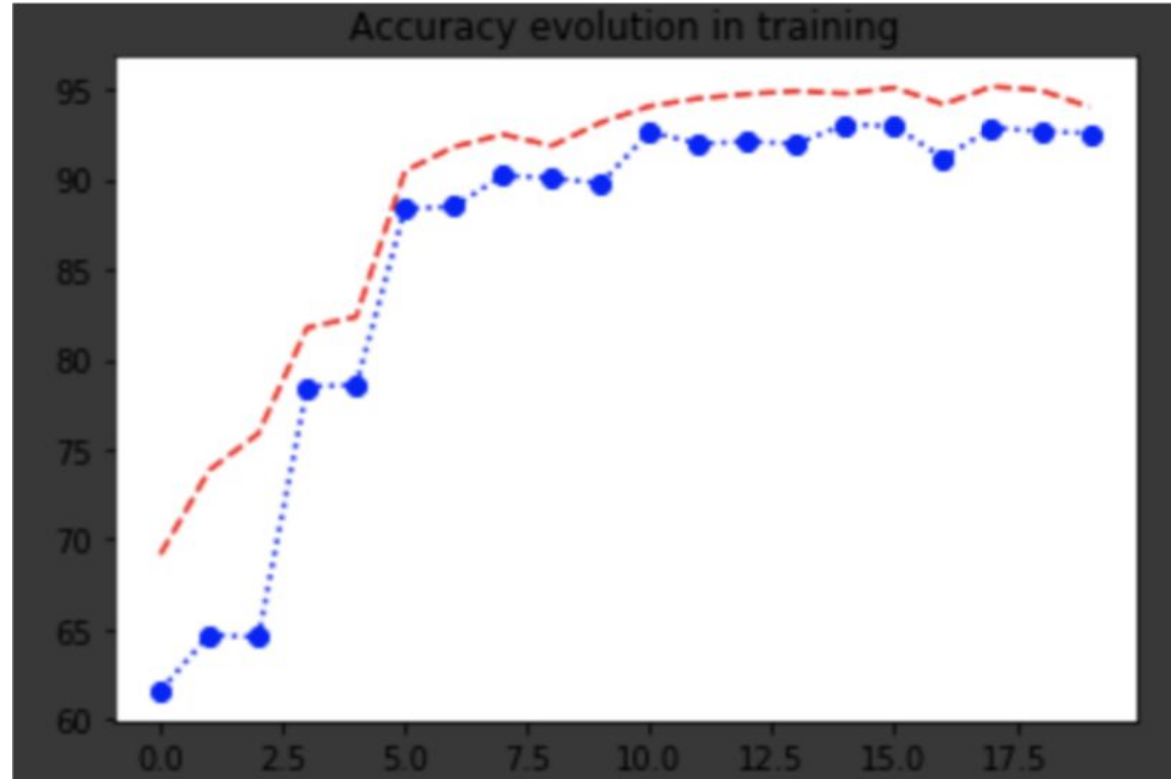
- Réelle plus value dans l'augmentation du dataset par rotation

Augmentation	Optimiseur	ModelNet10
Oui	SGD	0.83
Oui	Adam	0.83*
Non	SGD	0.69
Non	Adam	0.81

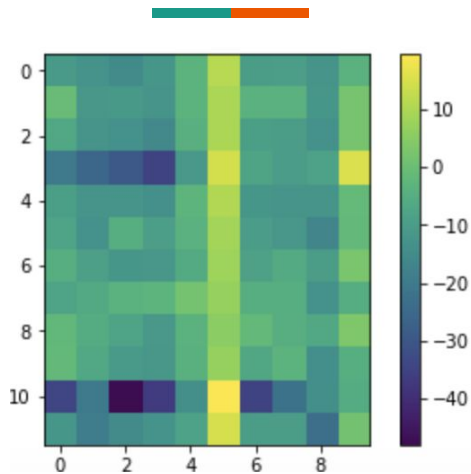
# Résultats



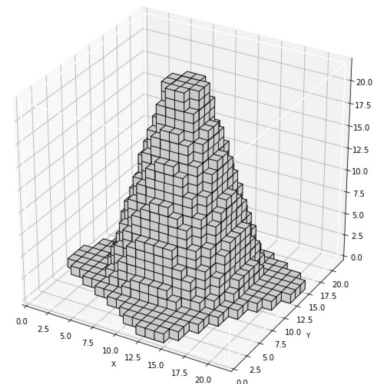
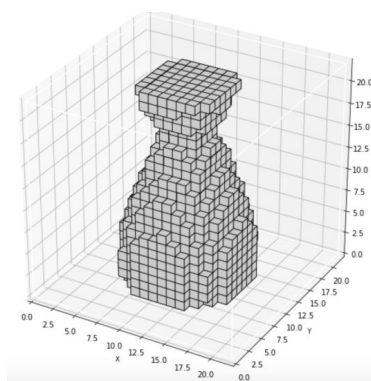
- Evolution de la précision en fonction du temps



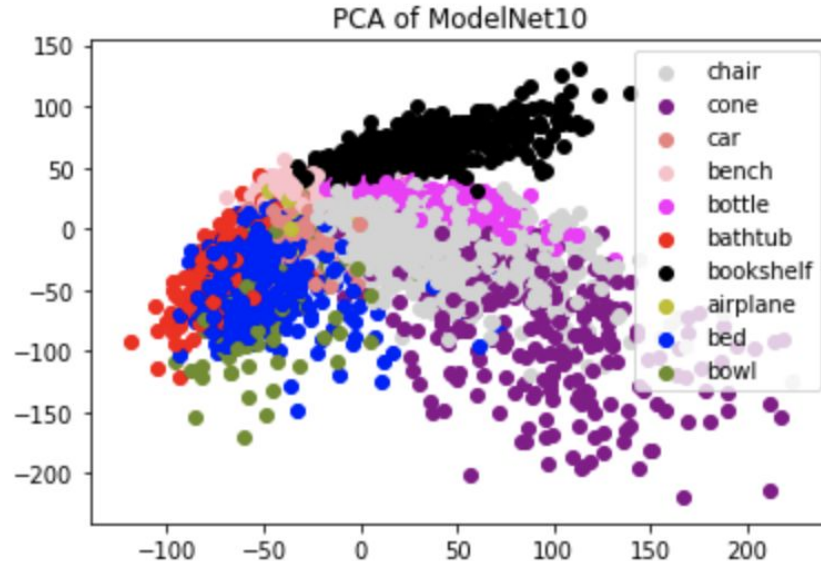
# Résultats : Invariance par rotation approchée



L'algorithme confond cette bouteille avec un plot. Surtout sa 4ème rotation. Mais encore une fois, il est capable de faire une bonne prédiction en considérant toutes les valeurs augmentées.



# Résultats : classification avec features de FC2



# Conclusion



- Réel apport de l'occupancy grid et du 3D-CNN
- Nous a permis de mettre en application un certain nombre de concepts du cours
- Difficulté de la gestion de la data !

# Extensions possibles



- Architecture ayant plus de 4 ans : Princeton ModelNet