1
Hello everyone, my name is Remi Cadene. I am a PhD student at LIP6, the Computer Science lab of the University Pierre and Marie Curie.
My Professors are Nicolas Thome and Matthieu Cord.
In this talk I will introduce efficient methods using CNNs to classify medium and small datasets.

2
The general context / in visual recognition is that / the amount of images and videos / is exponentially growing.
The classical methods to process visual data are the hand crafted models, such as using SIFT to extract features from regions of images, then using unsupervised techniques / to build bags of visual words, and finally, training Support Vector Machines.
Since 2 thousand twelve / CNNs appear to be the state of the art. The reasons behind this accuracy gap are the research efforts, the GPUs implementations and in particular the use of larger datasets. For instance, the well known Imagenet is made of 1.2 millions of training images and 1 thousand classes.

3
The thing is, we don't usually have access to this big amount of labeled images. In real life, everyone is not Google or Facebook.
In an internship I just completed, I studied three methods to tackle the classification problem in the context of low amount of images.
The first method consists of resetting all the parameters of a CNN designed for large dataset and to learn the parameters using Stochastic Gradient Descent.
The second method consists of using a pretrain CNN on Imagenet to extract features from the last layers and to use a SVM to classify the vectorized images.
The third method consists of resetting a certain amount of layers at the end of the network, to adapt the output size of the last layer to the new dataset and finally to learn all the parameters using SGD or Adam with a smaller learning rate for the pretrain layers.

4
We studied three CNNs of different input size, amount of parameters and depth. We also compared these networks by their forward and backward time. Notably using a CPU implementation can be 90 times slower than the best GPU implementation.

5
Let's see which methods to use to classify a medium dataset with a medium semantic gap with Imagenet, because you have images of food in the Imagenet dataset.
For this purpose, we will focus our experimental analysis on a dataset of food recipes called UPMC Food one O one. It is made of eighty thousand training images.

6
In this table we compare the models and methods we defined earlier.
Firstly, From Scratch is better than Extraction. It means that learning all the representations From Scratch is better than using the representation learned on Imagenet, even if the semantic gap is not so huge.
Secondly, Fine Tuning is the best approach for both CNNs (Overfeat and Vgg16) and InceptionV3 achieves better accuracy.

7.
Now, we validated that Fine Tuning is a good approach for medium datasets. Let's see if it still the case on a small dataset with a large semantic gap with ImageNet.
For this purpose, we will focus our experimental analysis on a dataset of satellite sight of roofs made of 8 thousand images in 4 classes (north-south, east-west, flat, other).
This dataset was provided for the Data Science Game Online Challenge. It is a international competition for master students and PhD students. And this year, more than 450 students, including a bunch of kaggle grand masters, tackled this challenge. My team from UPMC reached the first place.

8.
Here we compare several methods.
Firstly, we can see that Extraction achieves better accuracy than Hand Crafted models, even if the semantic gap is huge.
Secondly, Fine Tuning of the same CNN achieves better accuracy.
An other interesting approach could be to design a custom CNN for this task. **However this method has way more hyper parameters to optimize using grid search, such as the depth or the number of parameters in each layers.**
Finally, an ensemble by majority vote of fine tuned InceptionV3 allowed us to achieve the best accuracy.

9.
As for now, we only saw datasets where objects of interest are located at the center of the image.
But if it is not the case, we want our model to develop a strong translation invariance, **and data augmentation could not be enough for small datasets.**
To understand how CNNs achieve translation invariance, let's consider a toy dataset of human body and other classes.
If we train our CNN on images where human body are located at the bottom left only, the filters will learn how to detect parts of a body such as eyes, nose and combine them to finally learn non spatial representations in the fully connected layers.
However, when we test our CNN on images where human body are located at the top left, a location never seen before, the filters will still detect the parts, but the fully connected layers will produce different non spatial representations, so that the model could fail to detect a human body in this area.

10.
To add more translation invariance into CNNs, we can use a Multi Instance Learning framework. In doing so, we no longer consider an image but a bag of visual regions.
WELDON, a model developed in my research team at LIP6, presented a few month ago in this meetup by Matthieu Cord, allows to efficiently process the visual regions of an image. A pretrain CNN can receive images of a bigger size than expected to produce a spatial map of classes appearance. Finally, a spatial aggregation layer is used to select the maximum and minimum scoring regions to compute the final output.

11.
For the purpose of testing spatial invariance of CNN. We will focus our experimental analysis on a dataset of scenes made of 5 thousand training images for 67 classes.
Firstly, Fine Tuning achieves almost the same accuracy than Extraction, thus it is not able to adapt the representations for this task.
Secondly, WELDON achieves way better results than Fine Tuning and Extraction.

12.
To conclude, in this talk we explained that Fine Tuning + Ensemble is the approach of choice on medium an small datasets. Also, when a strong spatial invariance is required, a Multi Instance Learning approach, such as WELDON, can achieve way better accuracy.
However, beside our intuitions, we need theoretical proofs to deeply understand how CNNs converge and avoid overfitting.
Lastly, if you want to reproduce the experiments. You can use a library I designed for vision. It works as a plugin for the high level library for Torch called Torchnet. For instance, you can easily download and use the last pertain models and also iterate with parallelized data augmentation over a few datasets such as Imagenet, MIT67, UPMC-Food101.

Thank you for your attention.