## Problem Statement

The requirements of this Thesis were chosen from each Department I’m working with. TensorFlow was chosen. Computer Architecture Department and ImageNET Large Scale Visual Recognition Challenge VID challenge [43] from UPC Image Processing Group[23]. I will highlight the main motivations on these two decisions.

### Tensorflow



Figure 1.1: TensorFlow Logo.

Released on the 15th of November 2015 by Google[1], TensorFlow [21] is the newest open source library written in Python for numerical computation.

It has immediately a great success in the Machine Learning community and in less than one year it also had a lot of support and development by Google [1] itself, more over by many community projects, developed in any area of Deep Learning.

The peculiarity of TensorFlow is its work flux, made by data flow graphs. Where Nodes represent mathematical operations, edges represent the multidimensional data arrays communicated between them; the latter can be considered, as in elec- tronics, a Tensor, from here its name. In May 2016, Google [53] has revealed that it has used TensorFlow in AlphaGo project, with a special hardware dedicated to boost library’s performances.

To reach rapidly this goals, Google dedicated a special attention to the user experience of TensorFlow[21], which arrives with a great basic support and a well- grown GitHub community [22], the key of this quick improvement.

All these are the peculiarities that pushed us to choose TensorFlow [21], over more developed and bigger communities. The irruption made in the Deep Learning environment, shows up its great future potentialities, that are rolling out day by day.

### ImageNet



Figure 1.2: ImageNET Logo.

In its 7th edition, The ImageNet Large Scale Visual Recognition Competition (ILSVRC)[43] was a worldwide competition , where one of the 5 main challenges was VID. This has the aim of Tracking Object in Video, identifying eventually multiple- objects, classifying them and more over giving a track number for each of them, if it’s possible.

All the datasets (Training, Validation and Tests) are retrieved on the ImageNET Database, a collection of annotated pictures of different classes of objects. In the further chapter, I will explain in detail.

Developing a deep learning model in any area means to learn from the community and using, adapting, merging, writing code from it. So it was very difficult to reach the goal. TensorFlow community [22] has a fast growth, but the complexity of the models for the Object Tracking in Video are pretty far from the ones present in the nowadays community. Starting from the great idea of a model, composed by different parts proposed by K. Kang et al [32], which achieves the State of the Art in one of the categories of the last year competition, I used, composed and adapted some available models to reach the one I was looking for.

## The infrastructure: Asterix

The Asterix server is part of the BSC-UPC NVIDIA GPU Center of Excellence. To better understand the infrastructure I used, I will explain a little its architecture and show its schema:



Figure 1.3: Asterix Topology schema.

The machine has a total of 24Gb of local computational power inter-node, divided into to two NUMA (Non-uniform memory access) Nodes. This design architecture is specific for multiprocessing work, where the inter-node memory is used to boost performance. In the two nodes there are 16 CPUs, divided in four core for node, each of them is a Intel(R) Xeon(R) E5620 @2.4 GHz; Each CPUs has three different level of Cache Memory, L3 of 12Mb, L2 of 256Kb and the first level is split into L1i and L1d, both 32Kb. Each NUMA node is supported by four graphic cards, in the figure card from 1 to 4. All of them as visible into the below picture are NVidia Tesla K40c of 12Gb of memory. The ”c” version of the K40 version is the

one with an active fan sink that the card uses to cold down itself. This lets the card be suitable for many more purposes than other versions.

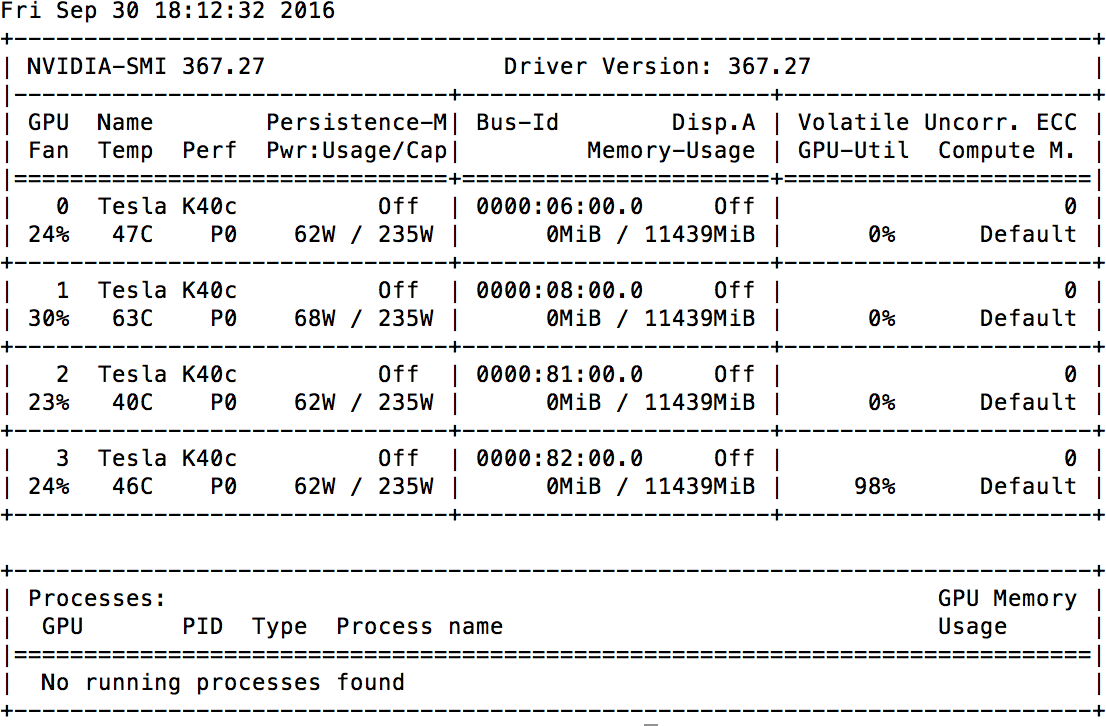


Figure 1.4: Terminal Output of the nvidia-smi command on the server.

In total to the 24Gb CPU memory are accompanied 48Gb of graphics card. These latter are important and fundamental for the Deep Learning topic. Because they are the ones that architecturally support naturally the mathematical computation.

The time constraints and the many problems and errors I encountered obliged me to learn, use and implement all the knowledge I needed for the topic in the fastest possible way. At the end of the Thesis[13], I will be able to respond to the question: can TensorFlow [21] be the fastest adaptable environment for Machine Learning, for implementing Research and Development, into a different infrastructure architecture with a great improvement perspective?

# Chapter 2

**State of the Art**

In this section, I will talk about the basics State of the Art for Visual Recognition. The concepts will be divided in the so called ”Still Image Processing”, because Time and Space properties of the analyzed photos are not considered and ”In Space and Time Image Processing”, where those parameters are considered and learned from the model.

## Still Image Processing

The Still Image Processing is the area that focuses the learning in three different specific goals: Classification of objects in images, their Localization and finally their Detection.

As anticipated, all these tasks are learned by the model without considering Space and Time properties, because the starting hypothesis is that sequential photos are not correlated in time.

### Classification

The Classification task is intended as the Visual Recognition of all the possible subjects inside the picture in the descending order of confidence, see Figure 2.1 for examples.

This task is at the base of all the others, because it recognizes if there are objects or not and which classes belong to. This is the starting point of the human visual focus on scene. Respect to the state of the art and the ILSVRC [43] challenges, in 2011 Classification was coupled with localization in a separate challenge and then from 2013 replaced by it, because the improvement rate was flat.

In the classification topic, we can cite many different models in a quality ascend- ing order, that corresponds also to the incremental evolution of them:

AlexNet (ImageNet Classification with Deep Convolutional Neural Networks, 2012) [33].

*•*

VGG-Net (Very Deep Convolutional Networks for Large-Scale Visual Recog- nition, 2015) [42];

*•*

* + - * GoogLeNet (2015) [47];

26

Batch Normalization, Google ( Batch Normalization: Accelerating Deep Net- work Training by Reducing Internal Covariate Shift) [30];

*•*

Deep Residual Learning, Microsoft ( Deep Residual Learning for Image Recog- nition)[25];

*•*

PReLu/Weight Initialization, Microsoft ( Delving Deep into Rectifiers: Sur- passing Human-Level Performance on ImageNet Classification)[26];

*•*

As we can extrapolate from the paper of Kaiming He et al.[26], a comparison of the five top methods, evaluated on the test set, is of the following order of magnitude:

|  |  |
| --- | --- |
| **Method** | **Top 5-err (test)** |
| **ResNet (ILSVRC ’15) [25]** | **3.57** |
| **BN-Inception [30]** | 4.82 |
| **PReLU-ne [26]** | 4.94 |
| **GoogleNet (ILSVRC ’14) [47]** | 6.66 |
| **VGG (ILSVRC ’14) [42]** | 7.32 |

Table 2.1: Comparison of the five top classification models on test set of ILSVRC.

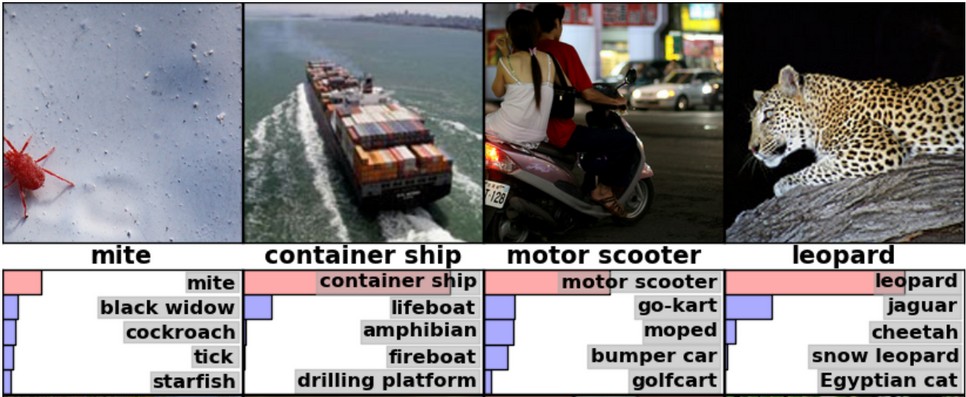


Figure 2.1: Inception model v3 examples of Image Classification (Recognition).

Due to the selected environment and its more available support, I chose the Inception model [20], that is now at the third version and it is fully implemented and supported by TensorFlow authors themselves [21]. This was one of the first decisions I made during the development of the work, sacrificing accuracy in order to respect time constraints and saving time to solve all the possible problems.

This model as we will see, is an evolution of the BN-Inception, that tries to max- imize the accuracy minimizing the workload effort. Due to convolutional networks that are at the base of most of the art models’ states, there is a lose of computational efficiency which could still be gained with factorized convolutions and aggressive reg- ularization to work on mobile vision scenarios. The structure of the Inception Model and the concepts behind it are based on four simple design principles, which improve the model computation efficiency and accuracy, as the paper from Christian Szegedy et al. [48], shows:

Avoid early bottlenecks in the model, let the acyclic graph be easily represented with a feed forward model from input to the regressor or classifier. More

*•*

important, avoid dropping big dimensions, and do it gradually from input to output, ensure not to loose useful correlations values for the task.

We can obtain a faster train by the use of high dimensions representations and by the activations per tile’s increase in the convolutional network.

*•*

To boost accuracy, it’s better to reduce dimensions before parameters aggre- gations; this due to the high correlation between adjacent units.

*•*

More the model is balanced and more the learning is consistent. So it’s impor- tant to raise in parallel width and depth of the network adding filters where needed.

*•*

This concepts applied to the original BN-Inception from Sergey Ioffe and Chris- tian Szegedy [30], that was itself an evolution of GoogLeNet by Christian Szegedy et al. [47], results in the following model structure:

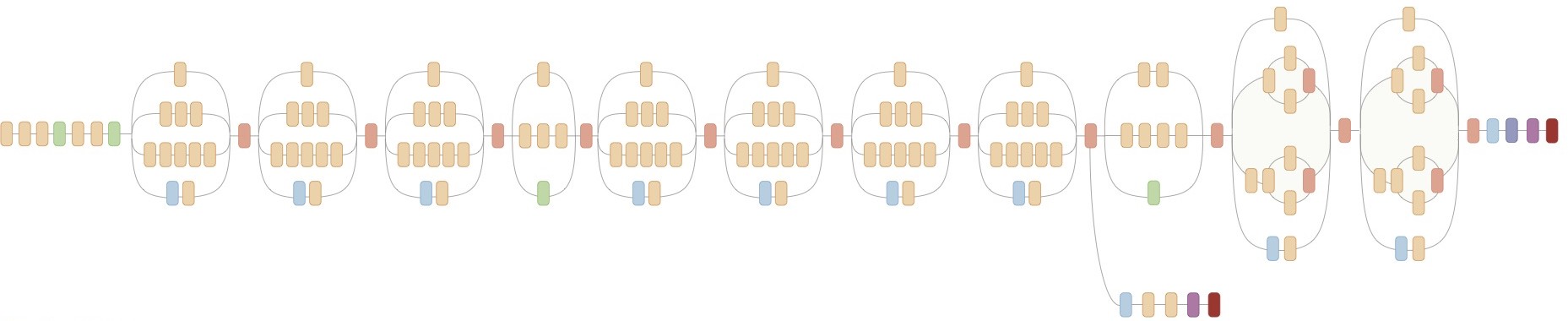


Figure 2.2: Architecture Representation of the Inception model v3 [48] developed by Google .



Figure 2.3: Legend of the different layers color, corresponding to the ones in the architecture representation of the Inception model v3.

As we can see from the figure 2.2, the graph is balanced in width and depth. It’s mostly composed by Convolutional Layers (the Figure 2.3, shows the layer colours legend), that make computation and extract features arrays of the images, then at the end of each architectural tier, they are averaged and concatenated with the following tier. It’s only at the two outputs of the model where Dropout and Fully- Connected tiers are used, linking layers, minimizing bottlenecks and sharing infor- mation between them and finally fading in Softmax layer to wonder the class. In term of evaluation of the models, in order to better understand the improvement rate we can look at the table below:

At the end, the starting decision I took brought me not to suffer in accuracy and save a lot of time instead of develop a basic component useful for detection and localization. I had only a nominal worst accuracy of 0.01 for Inception v3 respect

|  |  |
| --- | --- |
| **Method** | **Top 5-err (test)** |
| **Inception v3 (’15) [48]** | 3.58 |
| **BN-Inception [30]** | 4.9 |
| **GoogleNet (ILSVRC ’14) [47]** | 6.67 |

Table 2.2: Accuracy evolution of the inception model of Google from the first version, GoogleNet[47], to the third one [20] on test datset of ILSVRC 2012.

to the 3.57 of the ResNet. But the highest quality version of Inception, reaches 21,2% , top-1 and 5,6% top-5 error for single crop evaluation on the ILSVRC 2012 classification [28], setting the new state of the art. I think this is more than an acceptable and a great partial result of the project.

### Localization

Localization and Detection sound really similar and most of the community gets confused sometimes from the slightly difference. Both aim to detect from the input image to an output made of the presence/or not of objects. If they are present,they have to output an equivalent number of bounding boxes that highlight the objects in the picture.

But, the first one works with 1000 classes, still and moving objects, outputting the top 5-error labels it detects. Instead the second one works with only 200 moving objects typology and it outputs only the best one.

In few words, the difference is the typology of the objects, moving and still objects for Localization and only moving ones for the Detection.

Now that the difference is clear, I think it’s easier to understand why classifi- cation task was the principal starting challenge, from which they added a parallel Localization one, which they were merged in 2013 and in the same year a different task with less classes, Detection, and more other complex scenes and video challenges were added.

It is interesting to understand why I won’t go through models in this section but only with some other considerations, that will explain better the evolution of the challenges and of the topic itself.

In 2011 results [27] of the Classification plus Localization challenge there were only two brave participants, University of Amsterdam [2] & University of Trento and ISI lab. [51] of the University of Tokyo [51]. The winners of that year, Univer- sity of Amsterdam and Trento, were the first which implemented a selective search with some constraints, combined with an hierarchical grouping of adjacent regions of colour/text achieve a flat error of 0.425mAP score. This score is only 0.08mAP higher than the Classification result of the same participants for the only Classifica- tion category. 8% of increase is good but not so surprising. Above the best results from the winners model.

What is really clear from 2011 ILSVRC competition [43] is that Localization task could improve the accuracy of the classification, but it depends completely from the latter accuracy. This means that, the better is the Visual Classification of the image, understanding each ground, color and light level, with the scope to merge and combine those representations in the best way to achieve the highest ”human-level” knowledge, the better and easier will be the localization of the objects themselves.



The winners score gives a 50% overlapping error and a really higher accuracy depending on the classes. Comparing per-classes results of the classification and localization tasks, the paring was natural and a confirmation of the above concept. In Figure 2.5 some example of the overlapping error, related to images, where the subject is not thus evident.

So, it results clear that to boost accuracy is important to classify the image with different manipulations, size, crops etc. Obviously, this must be done in parallel with the growth of the length of the network, helping the algorithm to understand, also when it’s difficult for the picture, if which and where subjects are present in the picture.

This is not only a matter of algorithm but also, of the network experience, meaning a good train session, avoiding over-fitting, using the best datasets possible. These latter are the ones that define the quality source of the learning experience and it’s a concept never to underestimate.

From here forward, I will not differentiate between localization and detection again, because from my thesis point of view, results to be the same. To observe which dataset I was provided with, please read the Development of the Work Chap- ter. Then for a deeper knowledge about ImageNET datasets [40] and ILSVRC [43]competition, please, go to the Appendix 1.



Figure 2.5: Localization overlapping error examples, from the winners University of Amsterdam [2] & Trento [51], of ILSRVC LOC 2011[27].

### Detection

Detection, as discussed above, is the task to verify the presence of objects and if positive, highlight them with a bounding box. Previously, I talked about a selective search with hierarchical grouping of adjacent regions of colours. This was one of the first approach to solve that task, but now, I will, like for Classification, give a brief overview on the topic and than explain a little more in deep the one I choose.

Here are the main models for the detection topic:

* + - * Fast R-CNN, Microsoft Research (Ross Girshick)[14];

Faster R-CNN, Microsoft Research ( Faster R-CNN: Towards Real-Time Ob- ject Detection with Region Proposal, 2015)[39];

*•*

* + - * YOLO (You Only Look Once: Unified, Real-Time Object Detection, 2015)[38];

OverFeat, NYU (OverFeat: Integrated Recognition, Localization and Detec- tion using Convolutional Networks, 2014)[41].

*•*

This list of models touches all the types of basics strategies for the object detection. One of them the Faster R-CNN [39]could be also considered in the next paragraph, due to its naturally adaptation to the video task. But I will explain others models, more focused on achieving an In Time and Space learning model.

Here below a comparison of the models, to better understand the magnitude order relation: Fast [14] and Faster [39] R-CNN, are based on the R-CNN model

|  |  |  |
| --- | --- | --- |
| **Method** | **Test Dataset** | **mAP%** |
| **Fast R-CNN, [14] 2014** | VOC 07+12 | 68.4 |
| **Faster R-CNN (RPN + VGG, shared ), [39] 2014** | VOC 07+12 | 70.4 |
| **You Only Look Once, [38] 2015** | VOC 07+12 | 64.3 |
| **YOLO and Faster R-CNN, [38]2015** | VOC 07+12 | 75.0 |
| **Overfeat, [41] 2013** | ILSVRC2013 | 24.3 |

Table 2.3: Comparison of the main detection models, please pay attention to the different test database used and the implementation year.

from Ross B. Girshick et al. [15]. The Region Convolutional Neural Network, is a base architecture that analyzes the image at different level of convolutional and max pooling layers to produce a feature maps. Each output proposal goes through a Region Of Interest pooling layer that extracts fixed-length feature vectors. All this are faded into a Fully connected layer that outputs into two different softmax tier. The first one outputs the K object probabilities classes plus a catch-all “background” class. The latter returns four real values for each of the K item types, corresponding to the bounding boxes coordinates.

This model is no longer maintained, due to its successors, but in PASCAL VOC 2012 it was a state of the art, in which the authors used a variation with a bounding box regressor to boost performance to 53,3%mAP reaching a new limit. Here below a brief table resuming its historical results: In the following reference links, we can

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **VOC 2007** | **VOC 2010** | **VOC 2012** | **ILSVRC2013** |
| **R-CNN** | 54.2% mAP | 50.2% mAP | 49.6% mAP | - |
| **R-CNN bbox reg** | 58.5% mAP | 53.7% mAP | 53.3%mAP | 31.4% mAP |

Table 2.4: R-CNN median results, from the 2007 to the 2012 VOC dataset; the last column shows the mAP of the model on the ILSVRC 2013 dataset, this last result is more significant for my purposes.

see some per-class results, VOC 2007[15], VOC 2010[54] and VOC 2012[55]. The base R-CNN model is pretty slow to execute but also the Fast version that has some improvements. It’s the Faster version, introducing a smart tip, that can reach almost real time executions. This strategy is the Region Proposal Network, which used in parallel with the Fast R-CNN [39], can output cost-free region proposals, sharing the full-image convolutional features detection layers between both systems. Faster R-CNN [39] runs between 5-17fps and 58-59,9% mAP that are near the last year, state of the art value 62,4% mAP by MSRA[25].

Here some results of the Faster R-CNN on the PASCAL VOC dataset, that can help to understand why it’s considered as a milestone for models in the Vi- sual Recognition history and why it’s at the base of the state of the art detector, written by the same authors Kaiming He et al. The one that I chose, due to a yet semi-working project I found, is Overfeat[41]. Which is An integrated system for Recognition, Localization and Detection based on Convolutional Layers. The authors, Pierre Sermanet et al, starting from the winners model of 2012 [33], inte- grating sliding windows, multiscale classification, a Regressor and combining their predictions achieve a fifth place with a 13,6% top 5-error rate in 2013 post compe- tition. Where they replaced part of the model with a deeper model respect the one

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Training Data** | **Test Data** | **mAP** | **ms/fig** |
| **Faster RCNN** | 2007 trainval | 2007 test | 69.9% | 198ms |
| **Faster RCNN** | 2007 trainval + 2012 trainval | 2007 test | 73.2% | 198ms |
| **Faster RCNN** | 2012 trainval | 2012 test | 67.0% | 198ms |
| **Faster RCNN** | 2007 trainval&test + 2012 trainval | 2012 test | 70.4% | 198ms |

Table 2.5: The Faster R-CNN model considered is the version with the VGG-16 as Visual Recognition base model. The training and test data belong to PASCAL VOC competition.

of the competition.

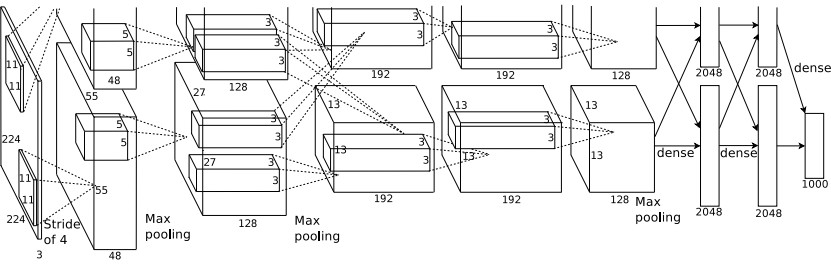


Figure 2.6: Schema of the best Classification model, from the winners of ILSRVC CLC 2012 [33].

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Layer | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Output 8 |
| Stage | conv + max | conv + max | conv | conv | conv + max | full | full | full |
| # channels | 96 | 256 | 512 | 1024 | 1024 | 3072 | 4096 | 1000 |
| Filter size | 11x11 | 5x5 | 3x3 | 3x3 | 3x3 | - | - | - |
| Conv. stride | 4x4 | 1x1 | 1x1 | 1x1 | 1x1 | - | - | - |
| Pooling size | 2x2 | 2x2 | - | - | 2x2 | - | - | - |
| Pooling stride | 2x2 | 2x2 | - | - | 2x2 | - | - | - |
| Zero-Padding size | - | - | 1x1x1x1 | 1x1x1x1 | 1x1x1x1 | - | - | - |
| Spatial input size | 231x231 | 24x24 | 12x12 | 12x12 | 12x12 | 6x6 | 1x1 | 1x1 |

Table 2.6: Architecture specification of the fast model used by the team during the ILSVRC CLC, LOC and DET 2013 competition

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Layer | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Output 9 |
| Stage | conv + max | conv + max | conv | conv | conv | conv + max | full | full | full |
| # channels | 96 | 256 | 512 | 512 | 1024 | 1024 | 4096 | 4096 | 1000 |
| Filter size | 7x7 | 7x7 | 3x3 | 3x3 | 3x3 | 3x3 | - | - | - |
| Conv. stride | 2x2 | 1x1 | 1x1 | 1x1 | 1x1 | 1x1 | - | - | - |
| Pooling size | 3x3 | 2x2 | - | - | - | 3x3 | - | - | - |
| Pooling stride | 3x3 | 2x2 | - | - | - | 3x3 | - | - | - |
| Zero-Padding size | - | - | 1x1x1x1 | 1x1x1x1 | 1x1x1x1 | 1x1x1x1 | - | - | - |
| Spatial input size | 221x221 | 36x36 | 15x15 | 15x15 | 15x15 | 15x15 | 5x5 | 1x1 | 1x1 |

Table 2.7: Architecture specification of the deeper model used by the team during the post ILSVRC CLC, LOC and DET 2013 competitions to reach the upper-cited result.

As we can see from the tables 2.6 and 2.7, both versions are well balanced in depth and height and follow mainly the rules described in the previous section 2.1.1 for Inception v3.

The ConvNet when applied with a sliding fashion approach are inherit efficient as the author [41] underlines; because at test time, the higher workload, caused by bigger images, is confined to a small region and it becomes more computational useful when resources are limited.

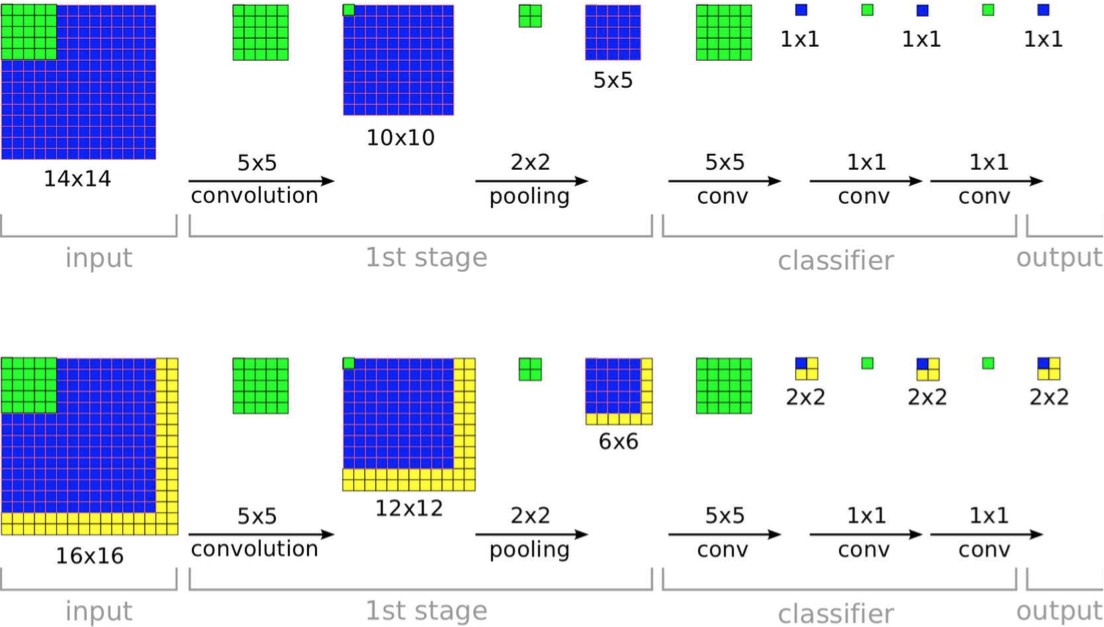


Figure 2.7: Schema of the ConvNet inherit advantages at run time.

The ConvNet use gives an hidden advantage to Overfeat[41], which is more computational aware of the resources to use; this is not to underestimate at all. In the Appendix 1 I will cite the YOLO[38] model I listed above, underling how its approach is one of the more innovative on the field; re-designing the main approach with a single neural network, instead of a deeper architecture.

## In Space and Time Image Processing

All the previous described task for Still Image Processing, can be combined with Time and Space constraints for video analysis. Single frames get correlated each others and the will to share information between them becomes immediate.

We are going to see the principals concepts behind these strategies and than some models that implement them, with more or less complex graph.

### Principals Concepts

The principal concept behind the In time and space analysis is the Slow and Steady Feature Analysis reported last year by the paper of Dinesh Jayaraman and Kristen Grauman [31]. This two analysis are based on the concept that, features, which are strictly correlated in time, have smooth changes. For a better comprehension look

at the Figure 2.8 that shows the schema of an optimizer, which has learned, how to plain changes in time and works as a smoother for functions in input.

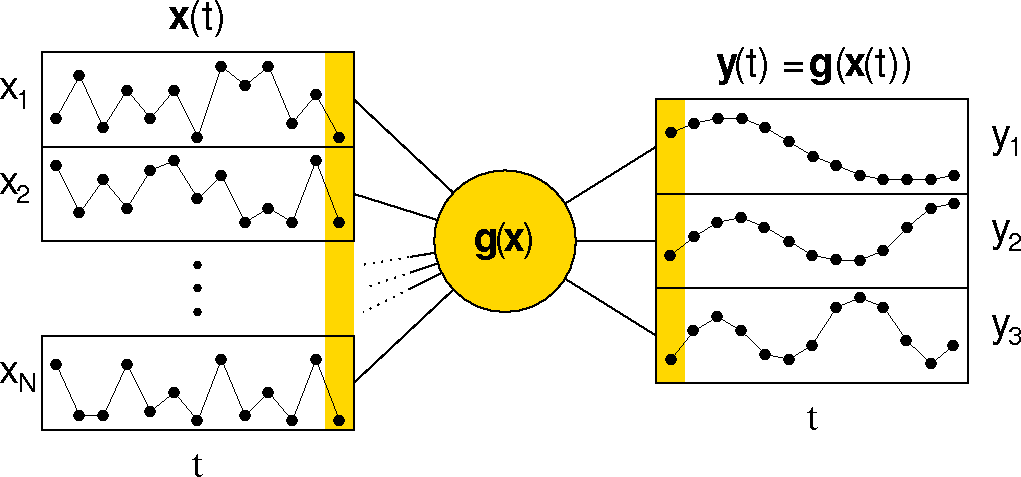


Figure 2.8: Concept Schema of a smoother unit for functions.

The use of this type of analysis at different levels of image features, lets the algorithm operate in time, smoothing changes between strictly correlated in time frames, but it also understands what is happening in the space dimension. This means that we have to work, not only with the first order of temporal coherence but also, with the higher ones. The best way to do it, is from the first frame to the last frame, following the entire scene, understanding all objects time and space’s properties. Here a schema showing the concept.

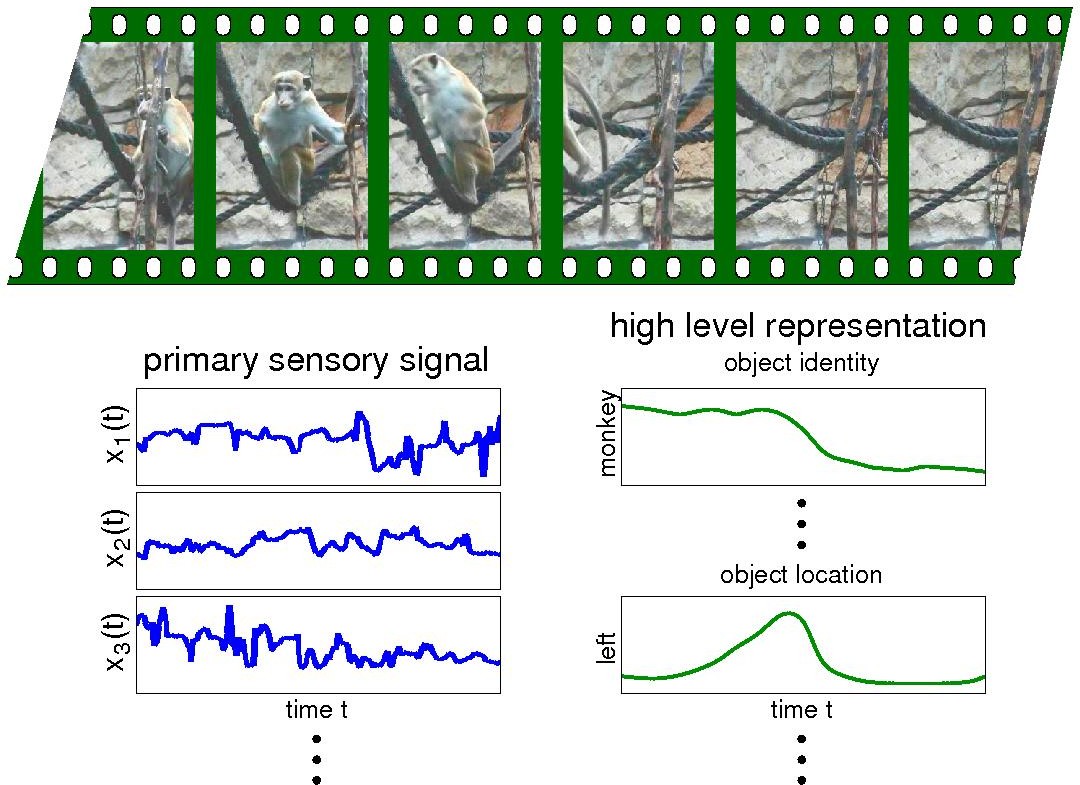


Figure 2.9: Concept Schema of a smoother applied to a Video.

The primary signals are represented under the picture on the left; instead on its Right, the high level representations of the smoothed extracted features show what the network is learning.

Using these concepts to develop an optimizer algorithm for the smoothing task an extracting from lower to higher levels features, it gives us the knowledge that: ”In the video there is a monkey that in the time and space dimension disappears from the scene”.