



---

## B31XP - Robotics Project

*Variational Image Clarification & Enhancement*

---

*Savinien Bonheur, Thomas Drevet & Omair Khalid*

*Supervised by : Alex Belyaev & Yvan Petillot*

---

*September/December 2017*

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Haze - Physical Model</b>	<b>3</b>
<b>3</b>	<b>State of the Art</b>	<b>5</b>
<b>4</b>	<b>Database Generation</b>	<b>7</b>
<b>5</b>	<b>The Fast Method</b>	<b>11</b>
5.1	Concept . . . . .	11
5.2	Implementation . . . . .	14
5.3	Results & Comparison with the State of the Art . . . . .	16
5.4	Limitations & Future Work . . . . .	18
<b>6</b>	<b>The Robust CNN Method</b>	<b>20</b>
6.1	Concept . . . . .	20
6.2	Implementation . . . . .	20
6.3	Results & Comparison with the State of the Art . . . . .	21
6.4	Limitations & Future Work . . . . .	23
<b>7</b>	<b>Conclusion</b>	<b>25</b>
7.1	Acknowledgement . . . . .	25
<b>8</b>	<b>Appendix</b>	<b>28</b>
8.1	Our methods on real images vs State of the Art . . . . .	28

# Chapter 1

## Introduction

This report details the work done for the Robotics Project (VIBOT - B31XP) module on the subject of 'Variational Image Clarification and Enhancement'. Image clarification and enhancement provides value in various applications such as underwater imaging enhancement, medical imaging enhancement, night imaging enhancement etc. The main focus of this project is single image dehazing, but tests have been conducted to check the validity of our approach for video dehazing. Dehazing is the process of removing haze from an image in order to enhance its visual quality and/or ease the information retrieval. For instance, edge extraction or any contrast based information retrieval are affected by haze.

Deep learning has become an attractive field of research in image processing, getting us outstanding results especially in object classification and recognition. Owing to its success and our strong interest in the field, a Deep Learning approach was chosen to spearhead our solution to the problem. In fact, Deep Learning has, in the past years, outperformed traditional approaches both in computation time and dehazing quality, according to the different publications on Deep Learning methods (presented later in the State of the Art).

This report constitutes the following: Chapter 2 defines haze and it's physical models; Chapter 3 provides our review of the literature; Chapter 4 concerns the creation of a database through the generation of synthetic non-homogeneous haze (having a non-constant density across the image), a problem which, in our knowledge, hasn't been studied by the literature; Chapter 5 explains in detail how, through Deep Learning, we achieved real time online dehazing, a first in the literature, while equalling the state of the art dehazing quality; Chapter 6 details how we identified and tried to achieve robustness against non-homogeneous haze (a problem which is yet to be tackled by the literature) by incorporating our real time CNN into a Multi-scale network (discontinuing online real-time performance); Chapter 7 presents the conclusion and acknowledgments.

# Chapter 2

## Haze - Physical Model

Haze is a phenomenon that appears when small particles obscure the sky and reduce its clarity [13]. Haze is a special case of Mie scattering where particles are much bigger than the incident wavelength, leading to non-selective, wavelength independent scattering (Figure 2.1).

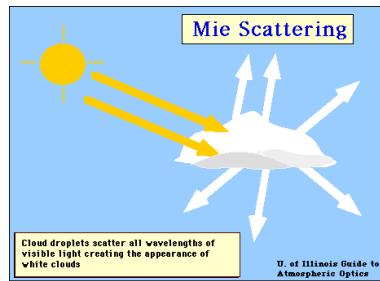


Figure 2.1: Non-selective Mie Scattering

The origin of the particles forming the haze is either meteorological e.g. water particles, spores etc. or human activity/pollution. In figure 2.2, we can see the effects of haze on image clarity due to the aforementioned sources.



(a) Meteorological origin



(b) Haze due to pollution

Figure 2.2: Haze Origins

Mie scattering describes haze on a microscopic level while our work concerns dealing with haze at a macroscopic level. Using Beer-Lambert law, the Atmospheric Scattering model (formula 2.1)

designed by McCartney [6] has been widely applied in the literature to describe the effect of haze. This model incorporates two parts: the attenuation of transmitted light  $t(x)$ , (formula 2.2), caused by the medium and the haze absorption ( $\beta$ ) multiplied by the distance between the light emitter and the measuring sensor ( $d(x)$ ); and the "Ambient light" ( $A$ ), formed by the scattering of the environmental illumination and linked to the quantity of light illuminating the scene.

$$I(x) = t(x)J(x) + A(1 - t(x)) \quad (2.1)$$

$$t(x) = e^{-\beta d(x)} \quad (2.2)$$

In this project, we only considered the haze affecting light in the wavelength range [ $\sim 400, \sim 800$ ] nm as it spans the human visible spectrum and consequentially, most of cameras' visible spectrum (this range might vary slightly depending on the camera sensor's quality).

# Chapter 3

## State of the Art

Utilizing Deep Learning for the dehazing application is something quite recent (first published method in 2016). Before this, the state of the art methods resorted to estimating the atmospheric scattering model parameters with the empirical Dark Channel Prior method [9] which states "in most of the non-sky patches, at least one color channel has very low intensity at some pixels" [9]. This prior is computed according to formula 3.1 and was initially used to estimate both the Ambient light ( $A$ ) by looking at the brightest pixel of the dark channel, and the channel wise transmission map ( $t(x)$ ), by using the dark channel darkest pixel patch wisely. Thanks to its success, the Dark channel is now widely used in method which are not using Deep Learning, within energies minimization framework.

$$J_{dark}(x) = \min_{c \in [r,g,b]}(\min_{y \in \Omega(x)}(J^c(y))) \quad (3.1)$$

Other approaches use neither the Dark Channel Prior or Deep Learning e.g. EVID [1] and FVID [2] methods. Both these methods aim to restore the image by performing a variational enhancement of the image to remove haze. Enhanced Variational Image Dehazing (EVID) from 2015 tries to maximize the contrast in the image before refining the output image by controlling the gray values and the saturation. Fusion-Based Variational Image Dehazing (FVID) from 2016, is an evolution of EVID where they improve the saturation computation, giving results that are slightly better than the original method. Although these methods give high quality dehazed image, their main drawback is the high computation time.

Although they provide insight to understand the atmospheric scattering model, classical approaches will solely serve as references for our approach since we decided to use a CNN based approach.

The first publication approaching dehazing via CNN was 'Multi-Scale Convolutional Neural Network' (MSCNN), from 2016 [15]. The main idea is to combine a fine and a coarse scale network to extract the transition map  $t(x)$ , the  $A$  is then deduced from  $t(x)$ . Combining these two scales networks leads to better result than using one deep network. Their implementations uses a combination of max-pooling and up-sampling layers to reduce the linearity of the network (and so its capacity of modeling non linear system). Their work also demonstrates the potential of Deep Learning applied to dehazing. In fact, MSCNN shows itself as being both faster and better (qualitatively and quantitatively) than the classical approaches that use the Dark Channel Prior approach. Their multi-scale framework will be reused in one of our proposed approaches as well.

The second Deep Learning based dehazing algorithm is DehazeNet' [3], from 2016.

This method share similarities with the 'traditional' dehazing methods. The process to get the estimation of the transmission ( $t(x)$ ) is accomplished in 4 steps in a cascade fashion. First, a features extraction layer extract haze-relevant features (as the Dark Channel for instance). Secondly, a multi-scale mapping make the features scale-invariant. In the third step, a local extremum remove the noise. Finally a non-linear regression (BReLU) layer is applied to avoid colour saturated artefact in the reconstruction. As in MSCNN, the ambient light ( $A$ ), is estimated from the transmission map ( $t(x)$ ). Although introducing new concept and linking CNN based approaches with traditional approaches, this techniques seems limited by its closeness to classical approaches.

AOD-Net, from 2017 [4], is the newest Neural Network applied to Deep Learning and provide the best results in the literature.

Their idea is to train a CNN to evaluate both the transmission and the ambient light as one parameter  $K$ . By calculating  $A$  and  $t(x)$  all-together they are able to reduce the reconstruction error jointly. This  $K$  parameter is then used in an reorganized Atmospheric Scattering model (formula 3.2), to get the dehazed image. Due to its performance and speed, this algorithm was used as a base for our fast method.

$$J(x) = K(x)I(x) - K(x) + b \quad (3.2)$$

$$K(x) = \frac{\frac{1}{t(x)}(I(x) - A) + (A - b)}{I(x) - 1} \quad (3.3)$$

The last CNN driven method is 'DeepDive' [10], from 2017.

In a new fashion, the network computes directly the dehazed image through a shallow network and without the help of atmospheric scattering model. This leads to three main limitation. The absence of intermediate coefficient limit the use and development of anti-flickering algorithm for video dehazing. The size of the kernel they use leads to blurriness. As their network do direct pixel operation it can create artifacts. This last issue was addressed by joining the result of the Euclidean loss function with a Feature loss function. This solution will be reused in our AMSCN to guide the removal of artifacts introduced by non-homogeneous haze in dehazed pictures.

All those techniques (Deep Learning and "classical") share a common limitation: they perform poorly on hazy night images. This issue arise as the Atmospheric Scattering model suppose a near constant ambient light ( $A$ ). In hazy condition, any strong enough incoherent light sources (relatively to the intensity of the ambient light), will create a visible cone of light as photons are scattered by the particles in suspension in the air. As the image is dehazed, the cone of light will persist as an artifact. Some techniques and models such as the Yu and al approach this problem [17]. However we will not tackle this limitation and will leave it as a ground for future work.

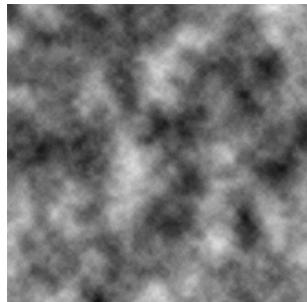
Following our study of the literature, we decided to implement two methods. A Fast dehazing CNN method, tailored for real time performances and inspired by AOD-Net, and non-homogeneous haze Multi-Scale CNN method, inspired by the MSCNN multi-scale architecture and DeepDive's two loss function.

## Chapter 4

# Database Generation

As there are no data-sets containing pairs of clear image and the corresponding hazy image (the light conditions and the camera pose might vary between a clear day and a hazy day) that are available online, there was a need to synthetically create such a data set for our use. To this end, the nature of haze was inspected by studying naturally hazy images, and certain deduction were made about what should be emulated by the synthetically created images.

Firstly, haze is non-homogeneous in nature, in that its concentration is not constant over space (e.g. the fog might be denser over a body of water due to its vaporisation). Although scene depth ( $d(x)$ ) certainly plays a major role in the amount of haze seen in an image, it is mathematically interchangeable with the absorption ( $\beta$ ). In fact, a haze twice as dense would look similar to a scene twice as far . As seen in Figure 4.1b, the amount of haze may vary significantly spatially i.e. two objects at the same distance may have different amounts of haze. To replicate the tropic nature of haze, Perlin noise is used 4.1a .



(a) Perlin noise model used for Haze Generation



(b) Example of Image with tropic Haze

Figure 4.1: tropic Haze

Secondly, the observed contribution of depth on an image haze is, solely, the increase of it i.e. the more the depth, the more the haze. As before, this means that we can emulate the effect of depth by simply increasing the amount of haze present (through  $\beta$ ). An observation of AOD-net field of view (a 13\*13 window) seems to indicate that the network doesn't estimate the depth while dehazing (as extracting the depth from a monocular camera would require a larger field of view). To validate this observation, we provided the MSCNN algorithm with a randomly hazed image before observing the extracted map. This map (Figure 4.2), despite the randomness of the haze, and thus, it's independence from the depth, stay strongly correlated with the haze. As dehazing is oblivious

to the depth, it is not required while generating data.

Our decision of ignoring depth information to generate dataset is further motivated by the lack of large or precise outdoor depth data-sets (e.g KITTI data-set is inaccurate for scene depth greater or equal to 80 meters).

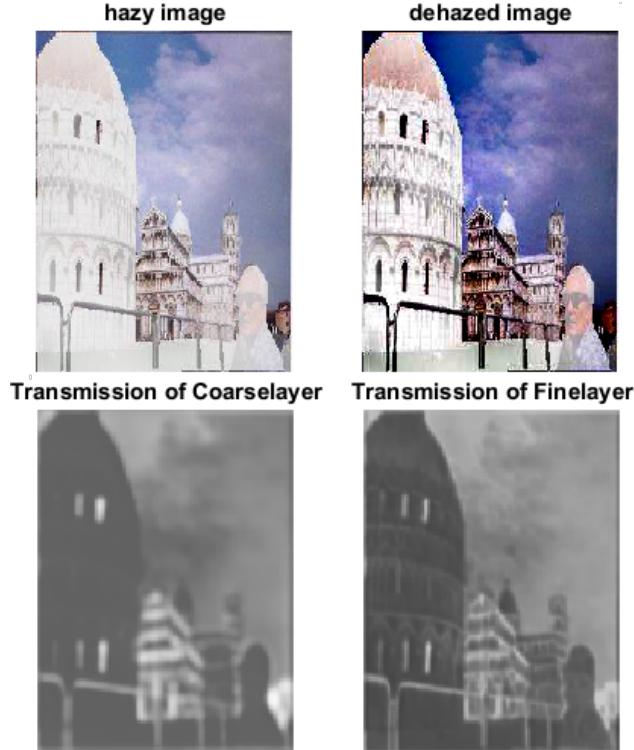


Figure 4.2: MSCNN transmission map for a non depth generated haze

Thirdly, the naturally hazy images may have sharper haze transitions due to adjacency of scene objects having a large difference of depth between them. In this case, the amount of haze present in adjacent patches will vary drastically. Figure 4.3a and Figure 4.3b help illustrate such an example. To incorporate this phenomenon, we shall resort to first dividing the image into a number of region and then treat each region as having a single  $\beta$  (Medium Absorption) value (more details below). Training a network with such data should provide robustness to complex shaped sharp haze transition.

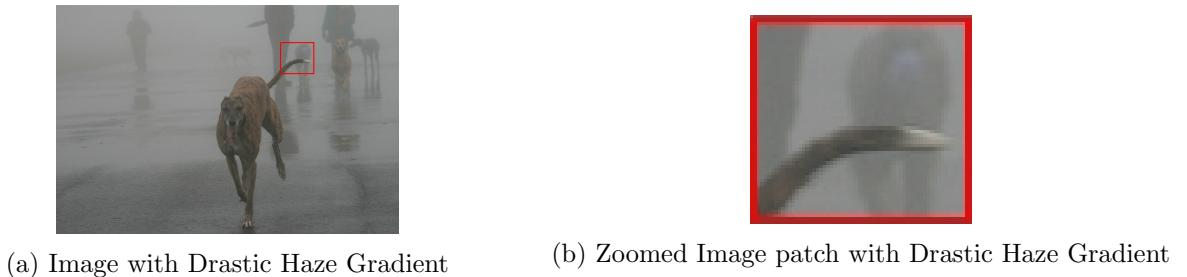


Figure 4.3: Example of Drastic Haze Gradient

Based on the previous observation we generate our dataset as follow:  
 We choosed the region segmented database SUN2012 Database [12]. SUN2012 Database contains 16,873 images and their segmentations. In order to speed up the training each images is resized to a [140,130] size. For each resized image, we assign to each segmented region an uniform random value  $\beta$  (different for each network, as their deazing capacity are different). We set, as a constant across the whole image, a global atmospheric lights  $A$  choosen uniformly in [0.5, 0.8]. Next, a randomized Perlin map is generated having the same size as the image, using four octaves and a frequency range of [10, 110]. These parameters are chosen empirically in order to create the most "haze-like" images

From the previous selected parameter, the  $t(x)$  is calculated as follows:

$$t(x) = e^{-\beta(x)*P(x)}$$

$$P(x) : \text{PerlinMap}$$

$$\beta(x) : \text{Medium Absorption Map}$$

The final hazy image is generated using the first equation 2.1 combined with our customized transmission. In Figure 4.4, some synthetically created images can be seen. Although the synthetic hazy images seem unnatural, they possess all the natural pattern of haze.



(a) Clear Images



(b) Synthetic Hazy Images

Figure 4.4: Clear-Hazy Image Pairs in Synthetic Hazy Data-set

# Chapter 5

## The Fast Method

### 5.1 Concept

To develop our Fast CNN dehazing method, we decided to analyse in depth AOD net internal operations.

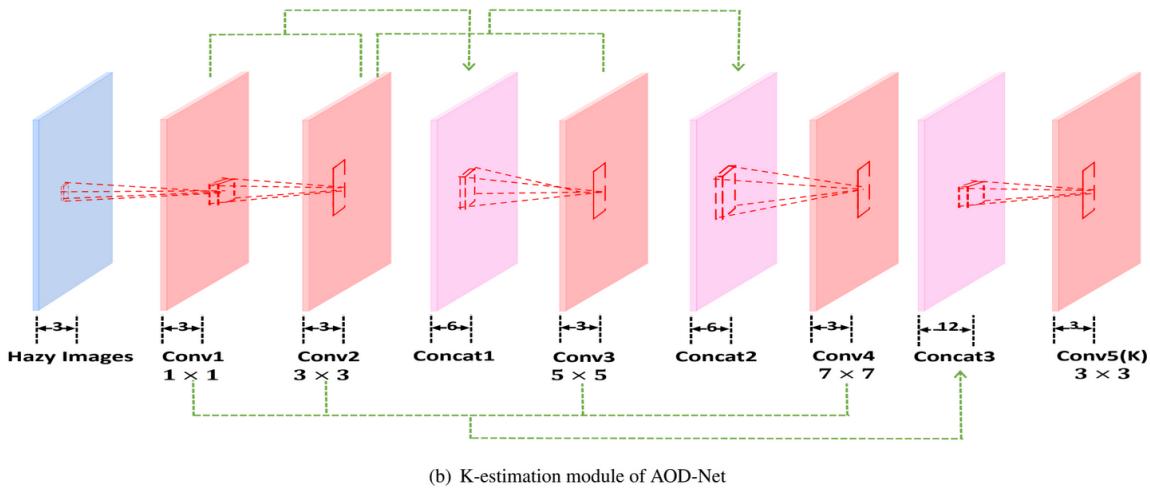


Figure 5.1: AOD-NET Design

As AOD-Net rely solely on the use of convolutional layers, we will recall how they works. In a neural network, each convolutional layer applies a kernel composed of  $n \times m \times d$  coefficients ( $c$ ) over its input before adding a constant bias. The result of those operations is then passed to an activation function to introduce non linearity. In this kernel,  $d$  is the depth of the input and  $n \times m$  the window size (for example  $3 \times 3$ ). To understand how each convolutional layer affect the network output, we will only consider the  $c$  coefficients as the bias is negligible for all the layers but the last one. To visualize each  $c$  coefficient contribution and, more generally, how each convolutional layer affects the extraction of the haze parameters  $K$ , we created the following graph: figure 5.2.

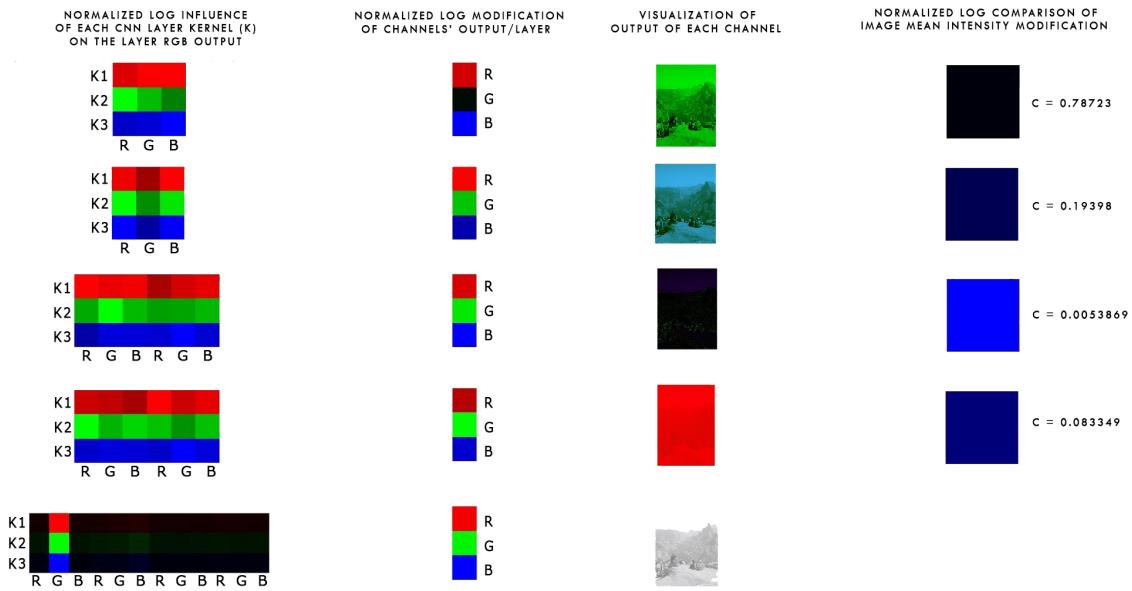


Figure 5.2: Study of the AOD-Net Layers

This graph is a quantitative representation of the AOD-Net internal operations. As AOD-net is compound of 5 convolutional layers, our graph presents 5 rows (one line for each layer). In this graph, each square represents a multiplying coefficient  $c$ . It is important to note than AOD net is only composed of  $c$  coefficients belonging to  $[-1;1]$  (with value close to 0 and  $\pm 1$ ). To help their visual comparison, we apply the following formula before display :

$$s = \log(\text{abs}(\frac{1}{c}))$$

. This formula inverses  $c$  in order to obtain a value of 1 for no transformation of the image mean and a value above 1 for every applied transformation. The presence of the absolute value is to allow the comparison of negative  $c$  and positive  $c$ . Lastly, the log reduce the dynamic range to avoid one  $c$  value darkening the other  $c$  value while displaying. In the first column are displayed each  $s$  coefficient (normalized by kernel) of each kernel by convolutional layer. As the output of each convolutional layer is always a 3 channels matrix, each line present 3 kernels (to help visualization, each kernel as a specific color). However, as some layers are fed with concatenated data, the number of row may vary between layers. In this column, the brighter a square, the more it will contribute to its kernel output

In the second column, in each line, are represented the  $s$  of the mean of each kernel  $c$  (the  $c$  are normalized per convolutional layer). The darker the square, the closer to 1 (compare to the two others kernel) will its  $c$  be. In application, as our  $c$  range between  $[0;1]$ , the darker a square, the more influential it will be for the convolutional layer.

The third column is the normalized output of each layer for a given picture . For example, on the first line, we can observe in column 2 that the second kernel (green) is darker (having a  $c$  closer to 1) than the 2 others kernel. As the red and blue channel are strongly attenuated (relatively to  $c$ ), the output appears green.

The last column represents the attenuation (normalized all together) of each input by layer. As we can see the third convolutional layer has the brightest square as its  $c$  is the smallest. It is important to note that  $c$  are independents to spatial relationships and thus doesn't capture in any mean the

pseudo derivative applied by the third layer.

From this graph we can extract two capital information. Firstly, the output layer (conv5) mainly rely on the output of the first layer (conv1) to calculate K (as in the first column, line five, only the green channel of conv1 appear bright). Secondly, K is close to independent from the channel colors (as the picture line 5 appears white).

To test those observation, we used a modified version of AOD-Net where only the first layer and the last layer are kept (figure 5.3).

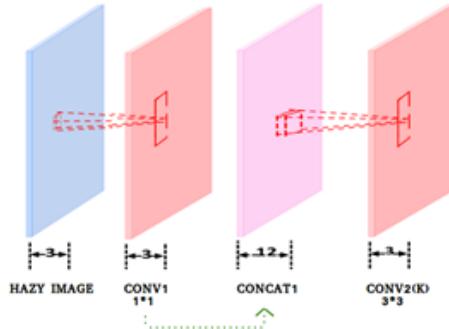


Figure 5.3: The truncated AOD-Net

As expected, this modified AOD-net achieve similar result (a mean PSNR of 49.69 out of 25 images, as you can see on figure 5.4) while reducing the computation time by a factor of 8 on a Intel i7-3630QM compared to the full AOD-Net net, for the dehazing time.

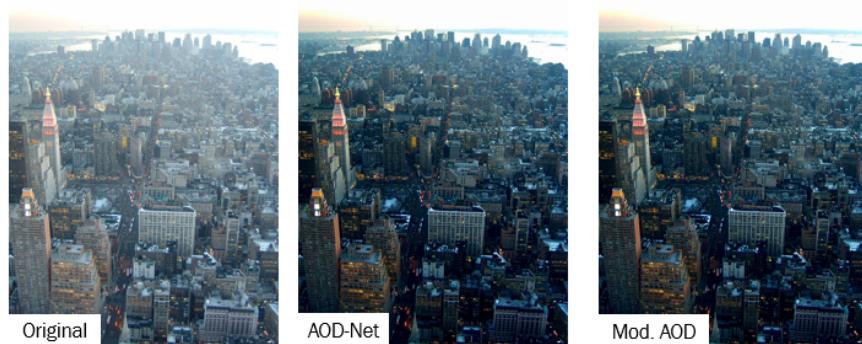


Figure 5.4: Results obtained by our modified AOD-Net

Remarkably, our truncated system, despite having a field of view of  $5 \times 5$  (3+3-1), perform as well as AOD-Net. This observation leads to the following conclusion: it is not necessary to evaluate the depth by any mean to dehaze a picture. As the depth is not used during the dehazing operation, the surrounding of a pixels should not influence it's dehazing. It should therefore be possible to dehaze with a field of view of  $1 \times 1$  instead of  $5 \times 5$ .

Based on the previous observations, we propose the following network :

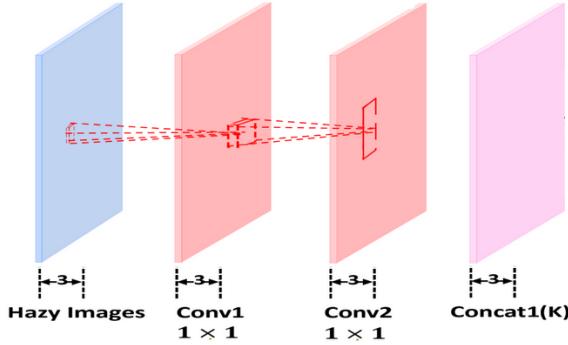


Figure 5.5: Our Fast CNN dehazing Design

## 5.2 Implementation

All our networks (Fast and Robust) are implemented using the Deep Learning framework Caffe[5]. Our training data will be a database generated as stated in chapter 4 with  $\beta$  ]. This choice was motivated by both the quantity of data available and the motivation of testing the validity of our database. The training of the fast network was accomplished using the following parameters :

- Solver SVD
- Batch size: 56
- Iteration number: 29000
- Learning Rate:  $10^{-7}$
- Steps size: 2500 iterations
- Learning Rate step multiplication factor for each step: 0.1
- Loss layer: Euclidian loss

Different parameters have been tested and some studies have been done to approach the optimal values. For instance, in figure 5.6 you can see the influence of different learning rate on the calculated loss.

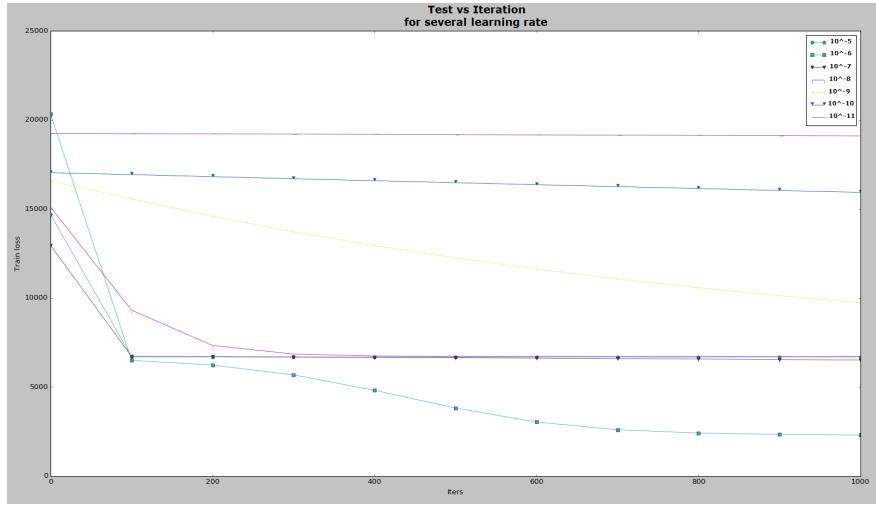


Figure 5.6: Learning rate study

As in ?? we clipped the gradient in order to constrain the norm within the norm  $[-0.1;0.1]$ . The influence of this clipping can be seen below.

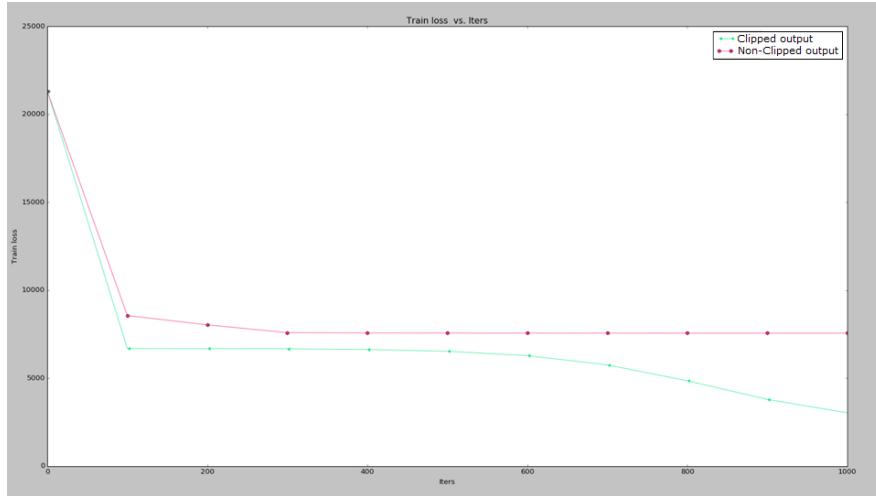


Figure 5.7: Clipped vs non-Clipped output

The size of the kernel was chosen to be  $1 \times 1$  to avoid any "white halo" effect. Indeed, a  $3 \times 3$  or bigger kernel would lead to some local averaging.

In the figure 5.8, we can see the training loss for all the training iterations and how fast the network converge. We obtain a convergence at around 2500 iterations. The training of our CNN took approximately 3 min on a Nvidia GEFORCE 740M. It can be noted than the simplicity of our network should preserve the Fast CNN from overfitting, i.e it should be possible to train the network with a really reduced batch size and a really small number of images (thus speeding the training furthermore).

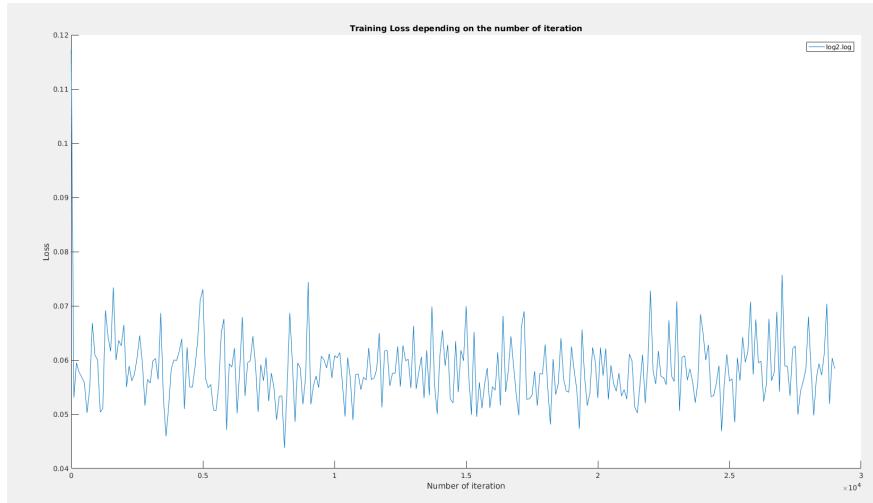


Figure 5.8: Training Losses during the training

It is important to note that the parameters chosen for the database generation, and thus the dehazing capacity of our network, were chosen to avoid saturation in low luminosity (thus loosing dehazing power). However a near AOD-net dehazing power (quantity of haze removed) could be achieved if the appropriate values were chosen while generating the database (however more dehazing leads to more saturation).

### 5.3 Results & Comparison with the State of the Art

In the following, we will compare our implementation to some of the methods presented in the State of the Art section: DCP, EVID, FVID, MSCNN and AOD-Net (the two others, DehazeNet and DeepDive, don't provide any algorithm working on our systems). We decided to use the images from two databases: the first dataset is the FRIDA dataset (Foggy Road Image Dataset) [8], a dataset composed of synthetic images representing different road scenes under different haze, while the second dataset is generated in a similar fashion as in chapter 4 (with images unseen by the network). Those two databases where chosen as they provide the ground truth images required for PSNR (Peak Signal to Noise Ratio) computation. We will also show result from real synthetics images.

Our first study is simply a visual evaluation of the images. You can see on figure 5.9 output images from the FRIDA dataset and on figure 5.10 output from our generated dataset.

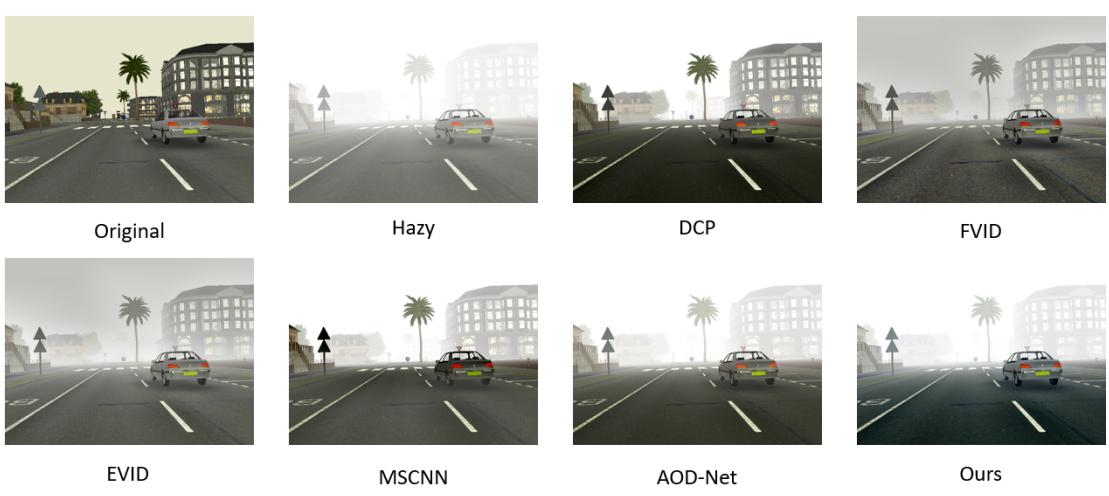


Figure 5.9: Comparison on the FRIDA dataset

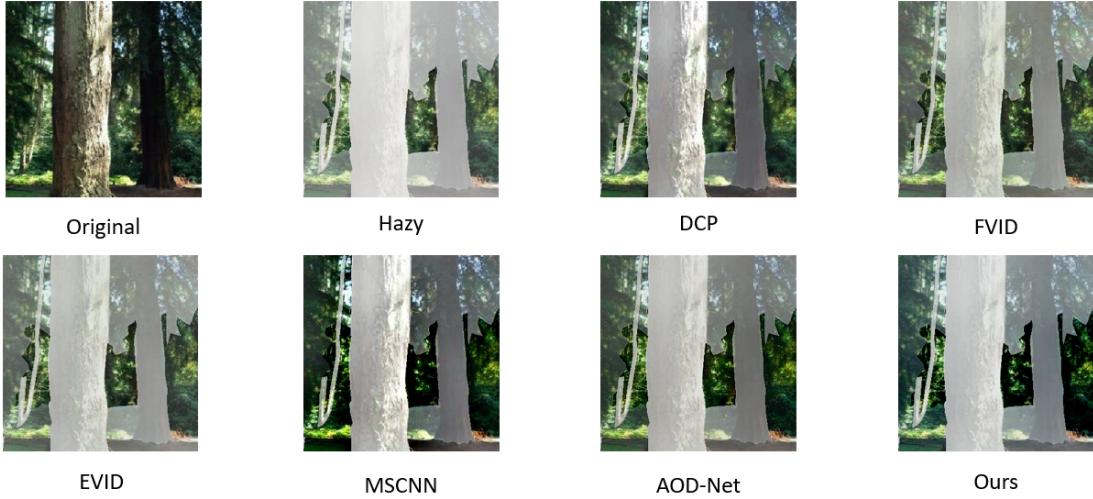


Figure 5.10: Comparison on our database

On the resulting images, our methods achieve a correct dehazing yet inferior to AOD-net. While FVID and MSCNN lead to the best results visually, they introduce visual artifacts under the form of white halo around dehazed object for the EVID algorithm , and as local saturation for the MSCNN algorithm. On the second dataset, we see that Fvid and Evid fail to perform. EVID (and thus FVID), minimize an energy function which seems to only hold for real looking hazy images. While MSCNN and DCP leads to the best result, our fast method still perform despite inferior results. Here is the board containing the mean of the images PSNR for all the methods, the FRIDA dataset (18 images) and our dataset (15 images).

Dataset	Noisy Image	DCP	EVID	FVID	MSCNN	AOD-Net	Ours
FRIDA	9.16	12.88	12.28	13.06	12.74	10.90	9.88
Ours	12.11	15.65	13.77	14.06	13.90	15.12	13.87

From this board, we can see the very poor PSNR results of our method to all the others methods, for both datasets.This could be explained, beside the relatively small size of the network, by the

low ambient light (i.e image luminosity) images used to train the network (see figure 5.11 for a low luminosity image dehazing). Noticeably, while MSCNN outperform visually the others methods on both datasets, it lead to over contrasted results on low luminosity images.



Figure 5.11: Our fast CNN applied on a image with low ambient light ( $A=0.5$ )

As you can see on figure 5.12, our tests on real images are encouraging too. In fact, despite increasing the image contrast, our results doesn't lead to artifacts and appear partially dehazed. Lastly, our algorithm takes only 0.012 s on a i7-3630QM to dehaze a 620X480 images.

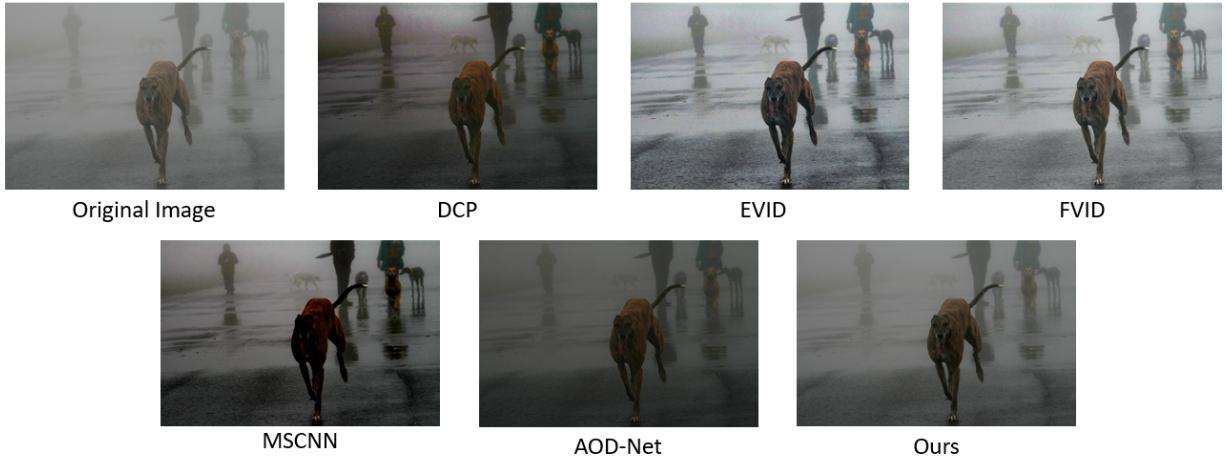


Figure 5.12: Comparison with real images

Interestingly, when tested on video, our fast network doesn't show any flickering effect.

## 5.4 Limitations & Future Work

Despite our decent results, there is few points that limits the performance of our Fast CNN dehazing method. Firstly, due to it's limited size (1), the network results are really sensible to the database generation parameters. For instance, our network, trained with a database containing too low ambient light images ( $A=[0.5,0.8]$ ) tend to perform remarkably better on similar low ambient

light condition FIG. Due to the limited size of the network and it's fast training, it could be interesting to develop an algorithm estimating the haze parameters ( $A$  and  $\beta$ ) in order to tailor the network training online for a given application. A whitening of the colors can be remarked on the network output. This whitening is induced by the color independent  $K$  output (one  $k$  shared by the three colors channels) of our network. As well, the low number of layers composing our network only provide a limited dehazing (as seen in the 'Result' section).

Our work on this method could be enhanced and tested in different situations: For instance, we haven't tried to do real-time dehazing on truly hazy condition (due to hardware limitation). In a similar fashion, it could be interesting to implement our fast method on hardware (such as a camera). Lastly, our method could be tested for underwater images enhancement by using a color dependant  $K$  output (one  $K$  for each color channel) as water diffusion is noticeably wavelength dependant.

# Chapter 6

## The Robust CNN Method

### 6.1 Concept

Our second task was the development of a second dehazing method overcoming the State of the Art (MSCNN, DehazeNet...) limitations against non-homogeneous haze. In fact, this problem leads us to build a new database of non-homogeneous hazy images (as explained previously in the report). Our idea is to adjoin a coarse scale network to our Fast CNN in order to weight it's dehazing. This type of multi-scale structure is used in the very well performing MSCNN process. Our network design is shown in the figure 6.1.

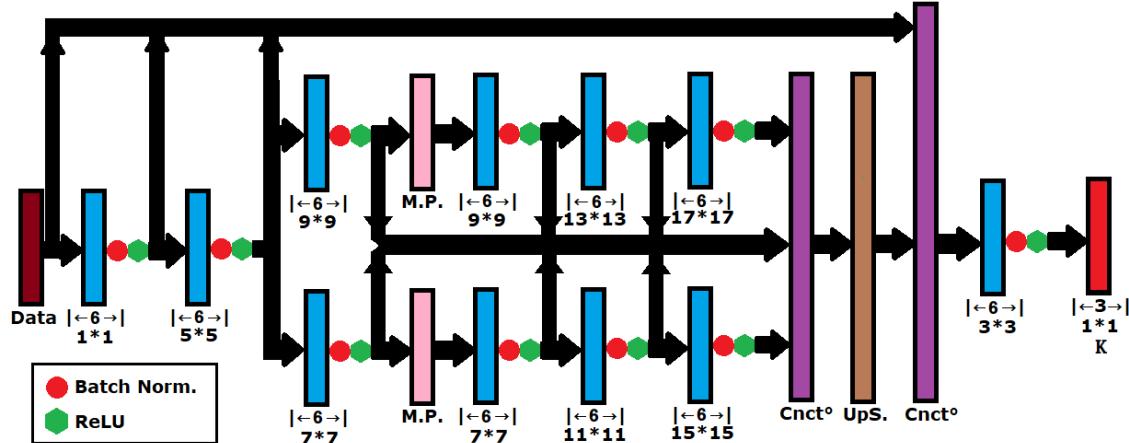


Figure 6.1: Our Robust CNN Design

To design it, we used the paper from X. Cao [16] in order to tailor the size of each layers (and kernels). This network is built with parallel process allowing a multi-scales analysis. However, this network is very redundant in its field of view.

### 6.2 Implementation

The training of the Robust network was accomplished using the following parameters:

- Solver: AdaDelta
- Batch size: 8

- Iteration number: 93000
- Loss layer: Euclidian loss

We restricted our batch size as well as the training iteration size due to time limitation. We can remark in figure 6.2 that 75 000 iterations present a lower error on the testing database (likely due to a too small batch size allowing too much variation between the training batches). However, results are more visually appealing after 93 000 iterations. This seems to indicate a still training network as well as it show the limitation of the Euclidian loss applied to dehazing quantification.

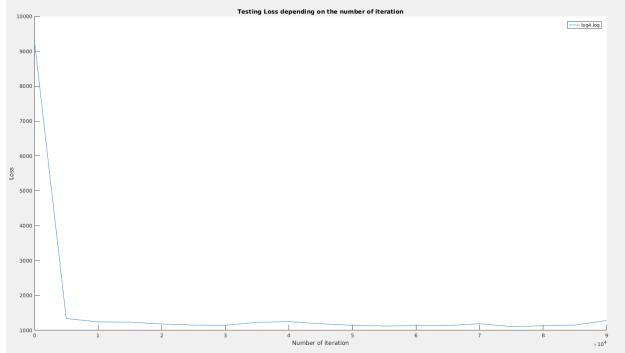


Figure 6.2: Robust network Testing vs Iteration

To generate the database, we once again proceed as Chapter 4 while setting  $\beta$  in the range of [4;8] and generated 11000 image as a training set and 4000 images as a testing set.

Our algorithm take 2.52s on a i7-3630QM to dehaze a 620X480 images.

### 6.3 Results & Comparison with the State of the Art

In this section we will compare our results using the same databases as before (DCP, EVID, FVID, MSCNN and AOD-Net). Despite a good PSNR, our results present a lot of artifacts while showing a close to MSCNN dehazing behaviour (therefore proposing a clearly worse dehazing quality). On figure 6.3, you can see the results we obtain on the FRIDA dataset and on figure 6.4, the results on our database. Meanwhile, the table try to quantify the dehazing quality on both datasets through a PSNR calculation.

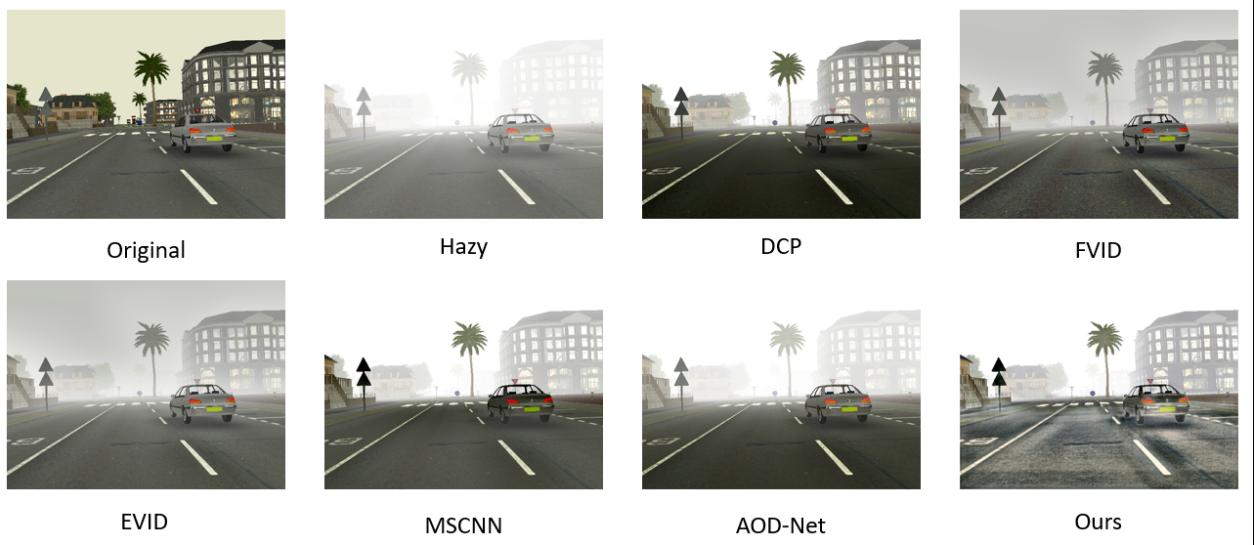


Figure 6.3: Comparison on the FRIDA dataset

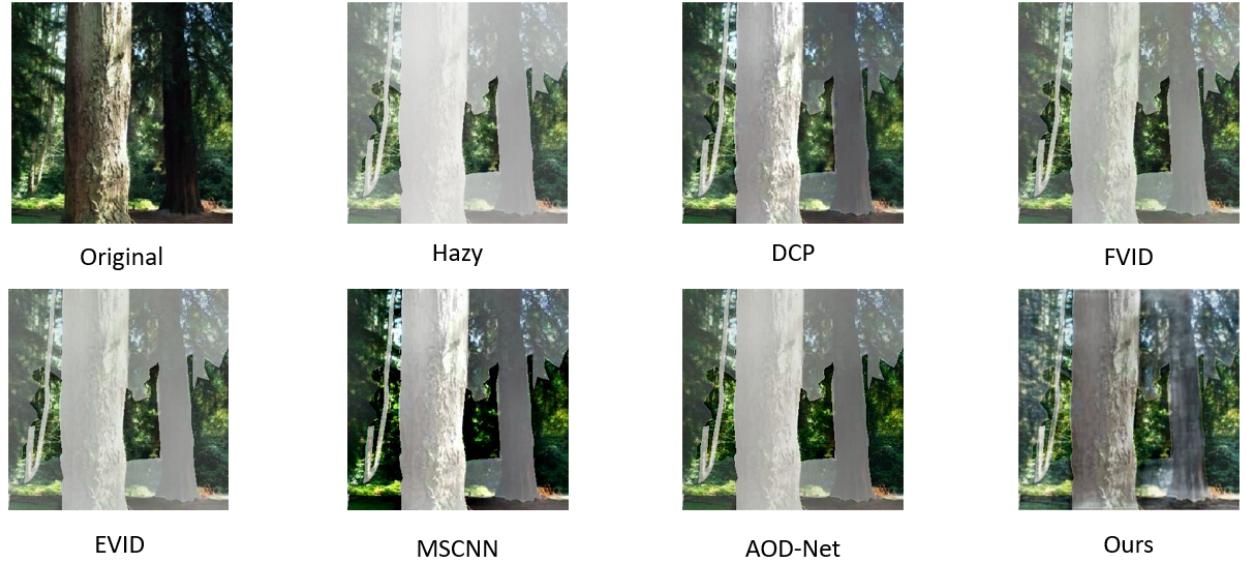


Figure 6.4: Comparison on our database

Dataset	Noisy Image	DCP	EVID	FVID	MSCNN	AOD-Net	Ours
FRIDA	9.16	12.88	12.28	13.06	12.74	10.90	12.37
Ours	12.11	15.65	13.77	14.06	13.90	15.12	13.8

From the results, some effects can be pointed out: first, as the DeepDive method, we can see some halos around dehazed objects due to the large kernel used for dehazing (border effect also appears on the edges of the image). We can also notice a general increase of the images brightness due to the lower A range ( $[0.5, 0.8]$ ) used by our method compare to AOD and MSCNN ( $A=[0.5, 1]$ ). Lastly, our network introduce non continuous hazy patches in parts of the images presenting light haze. We interpret this haze introduction from the network as an attempt to limit non homogeneous haze on the output images. Indeed, instead of solely removing haze, the network also seems to add haze in an attempt of providing a homogeneously hazy output image (therefore accomplishing partly

the wanted behaviours).

From the database pictures and the real images (figure 6.5), we observe encouraging results. Indeed, despite the artifacts cited previously, we are able to dehaze the images hazy patches much more than the literature methods (likely due to the extremely hazy dataset used to train the network).

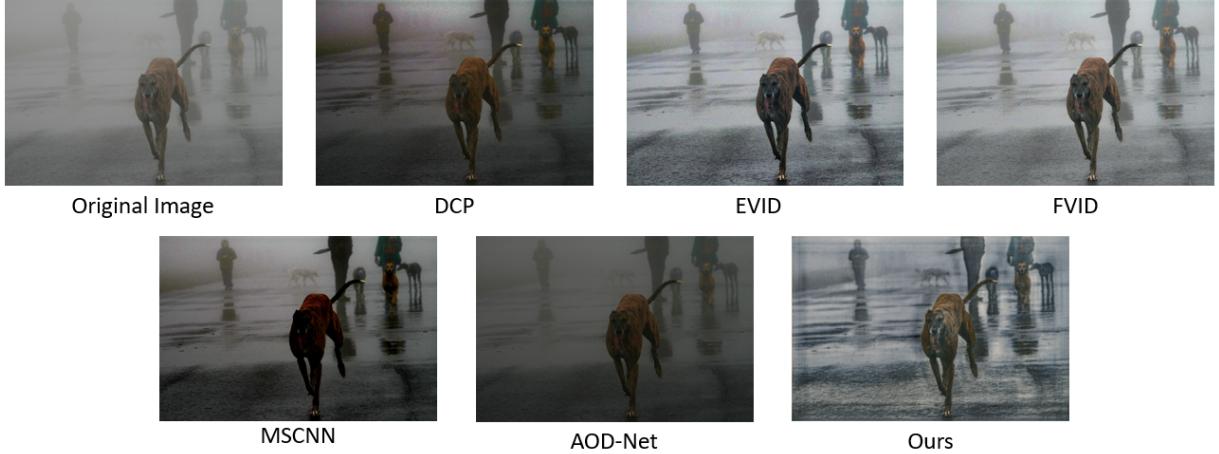


Figure 6.5: Comparison on real images



## 6.4 Limitations & Future Work

As stated previously our method present few artifacts (white halo, appearance of patches of haze,...). We believe a more rigorous training (more iterations, bigger batch and bigger training images) could reduce most of those effects. We also believe that adding another loss function penalizing features artifact (halo, non-homogeneous dehazing) would help speed up the training while diminishing artifacts. Through some tests (non present in this report due to time and concision limitation), we think that reducing the size of the kernel for one of the max pooled branch could

allow to overcome the halo effect. Finally, similarly as the fast implementation, this network (once corrected to limit artifacts) could be applied to other visual enhancing problem such water imaging enhancement.

# Chapter 7

## Conclusion

Those 4 months of project allowed us to apprehend and grasp the challenges of dehazing as well as the current developed solutions. After a shallow study of the existing state of the art approaches, both Deep Learning based (MSCNN, AOD-Net, DeepDive, ...) and more classical (Dark Channel Prior, FVID,...), we built two algorithm to overcome two identified limitations of the current State of the Art: a fast network to achieve real time dehazing, and a robust network to fully remove non-homogeneous haze. In order to achieve such results, we introduced a novel way of building databases leading to a considerable increase in the number of training images available. This database by allowing the training of two neural networks as well as by providing consistent PSNR testing results (only one method seems to perform significantly differently on our dataset) showed itself to be very promising (some test are still required to ensure the complete validity of such dataset). Trained from this dataset, our fast network, (designed as a simplified AOD-Net) is able to achieve real-time dehazing (a first in the literature to the extent of ours knowledges) despite showing clear dehazing limitation when compare to it's inspiration AOD-Net (mostly due to a poor choice of dataset parameters and possibly linked to an oversimplification of the two used layers). However the short training time as well as the limited dataset size required to train this very shallow network could lead to its application on well bounded hazy cases (instead of one network dehazing any hazy images, a on the fly trained network adapted to particular hazy condition). Trained from a similarly generated dataset, our robust implementation shows encouraging results. In fact, this network seems to try to equilibrate the haze on all the output images (thus showing the desired behaviour). However, while performing well on some images, this network still lead to artifacts (white halo around dehazed objects, border effect, random hazy patches). Overall, we are satisfied by the work accomplished during this period. While familiarizing ourself with Deep Learning (a cutting edge tool in which we are interested) and gaining insight about the challenges of dehazing, we built performing tools tackling the limitations of the current existing method (computation complexity, robustness to non uniform haze and the difficulty of building large dataset) that may serve as basis for later works.

### 7.1 Acknowledgement

We would like to thanks Pr. Alex Belyaev, our supervisor in this project, for his support and his advice and Pr. Yvan Petillot, our project lecturer, for his attention and his remarks. Finally, we want to say ‘thank you’ to our classmates for their helps and suggestions on our project.

# Bibliography

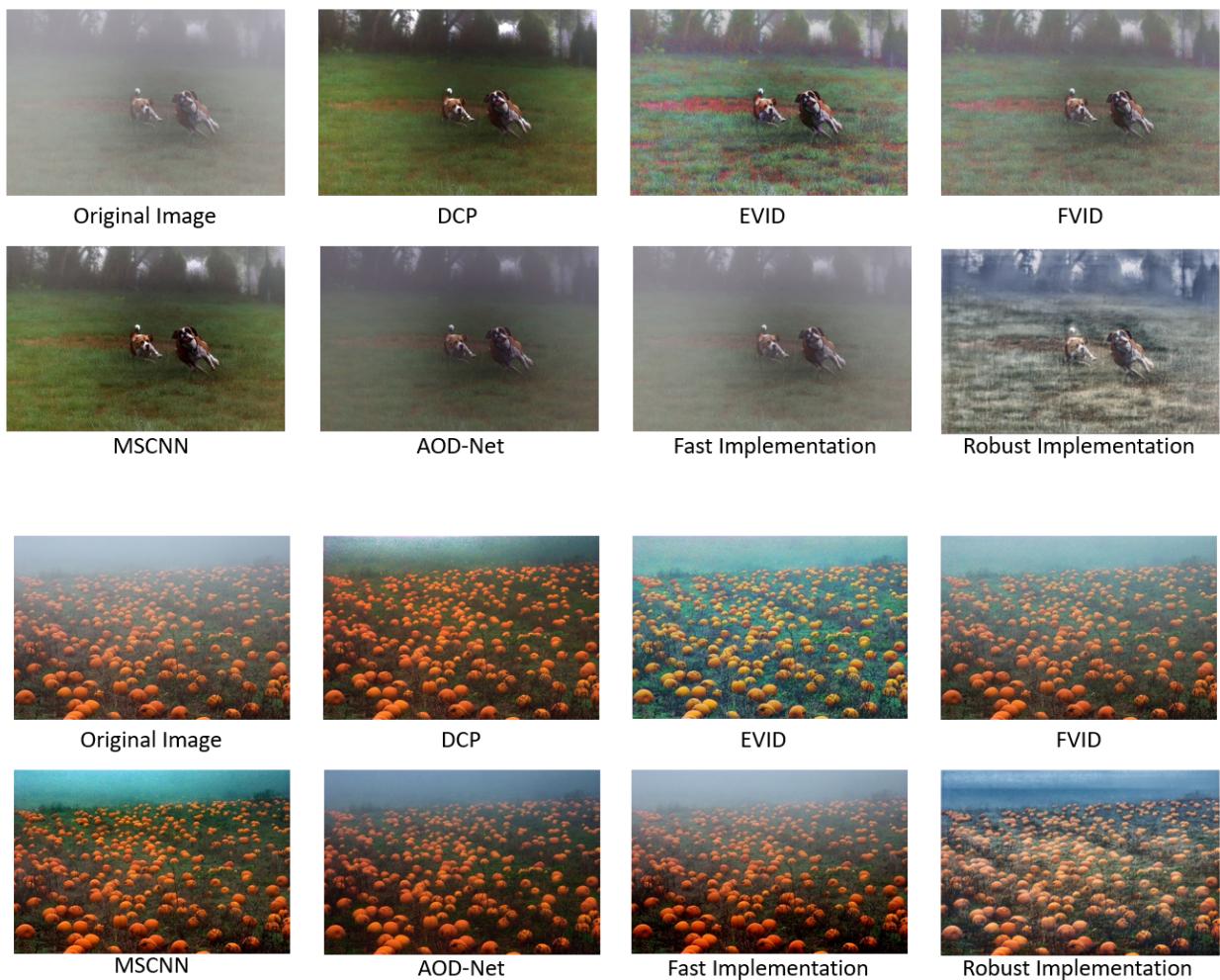
- [1] A. Galdran, J. Vasquez-Corral, D. Pardo, and M. Bertalmío *Enhanced Variational Image Dehazing*. SIAM Journal on Imaging Sciences, vol. 8, no. 3, pp. 1519–1546, 2015.
- [2] A. Galdran, J. Vasquez-Corral, D. Pardo, and M. Bertalmío *Fusion-Based Variational Image Dehazing*. IEEE Signal Processing Letters, 2016.
- [3] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. *DehazeNet: A End-to-End System for Single Image Haze Removal*.  
<https://arxiv.org/abs/1601.07661v2>
- [4] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng. *An All-in-One Network for Dehazing and Beyond*.  
<https://arxiv.org/abs/1707.06543>
- [5] *Caffe Deep Learning Framework*  
<http://caffe.berkeleyvision.org/>
- [6] E. J. McCartney *Optics of the Atmosphere: scattering by molecules and particles*. New York, John Wiley and Sons, Inc., vol. 1, 1976
- [7] Git. *DCP* <https://github.com/sjtrny/Dark-Channel-Haze-Removal>
- [8] J.-P. Tarel, N. Hautière, A. Cord, D. Gruyer and H. Halmaoui *Improved Visibility of Road Scene Images under Heterogeneous Fog* IEEE Intelligent Vehicles Symposium (IV'10), San Diego, CA, USA, June 21-24, 2010.
- [9] K. He, J. Sun, and X. Tang *Single Image Dehazing Removal Using Dark Channel Prior*. IEEE Trans Pattern Anal Mach Intell, 2011
- [10] L. Goncalves, J. Gaya, P. Drews-Jr, and S. Botelho. *DeepDive: A End-to-End Dehazing Method Using Deep Learning*. Graphics, Patterns and Images (SIBGRAPI), 2017 30th Conference on
- [11] R. Pascanu, T. Mikolov, and Y. Bengio. *On the difficulty of training recurrent neural networks*. ICML, 2013.
- [12] *SUN Database*  
<https://groups.csail.mit.edu/vision/SUN/>
- [13] Wikipedia. *Haze*.  
<https://en.wikipedia.org/wiki/Haze>
- [14] Wikipedia. *Mie*  
[https://en.wikipedia.org/wiki/Mie\\_scattering](https://en.wikipedia.org/wiki/Mie_scattering)

- [15] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. *Single Image Dehazing via Multi-Scale Convolutional Neural Networks*. European Conference on Computer Vision, pp. 154-169, 2006
- [16] X. Cao *A practical theory for designing very deep convolutional neural network* No Publication
- [17] Y. Li, R. T. Tan, and M.S. Brown *Nighttime haze removal with glow and multiply light colors*. ICCV, 2005

# Chapter 8

## Appendix

### 8.1 Our methods on real images vs State of the Art







Original Image



DCP



EVID



FVID



MSCNN



AOD-Net



Fast Implementation



Robust Implementation



Original Image



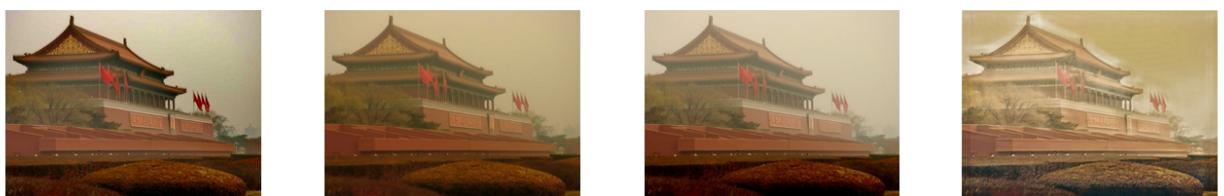
DCP



EVID



FVID



MSCNN



AOD-Net



Fast Implementation



Robust Implementation



Original Image



DCP



EVID



FVID



MSCNN



AOD-Net



Fast Implementation



Robust Implementation

