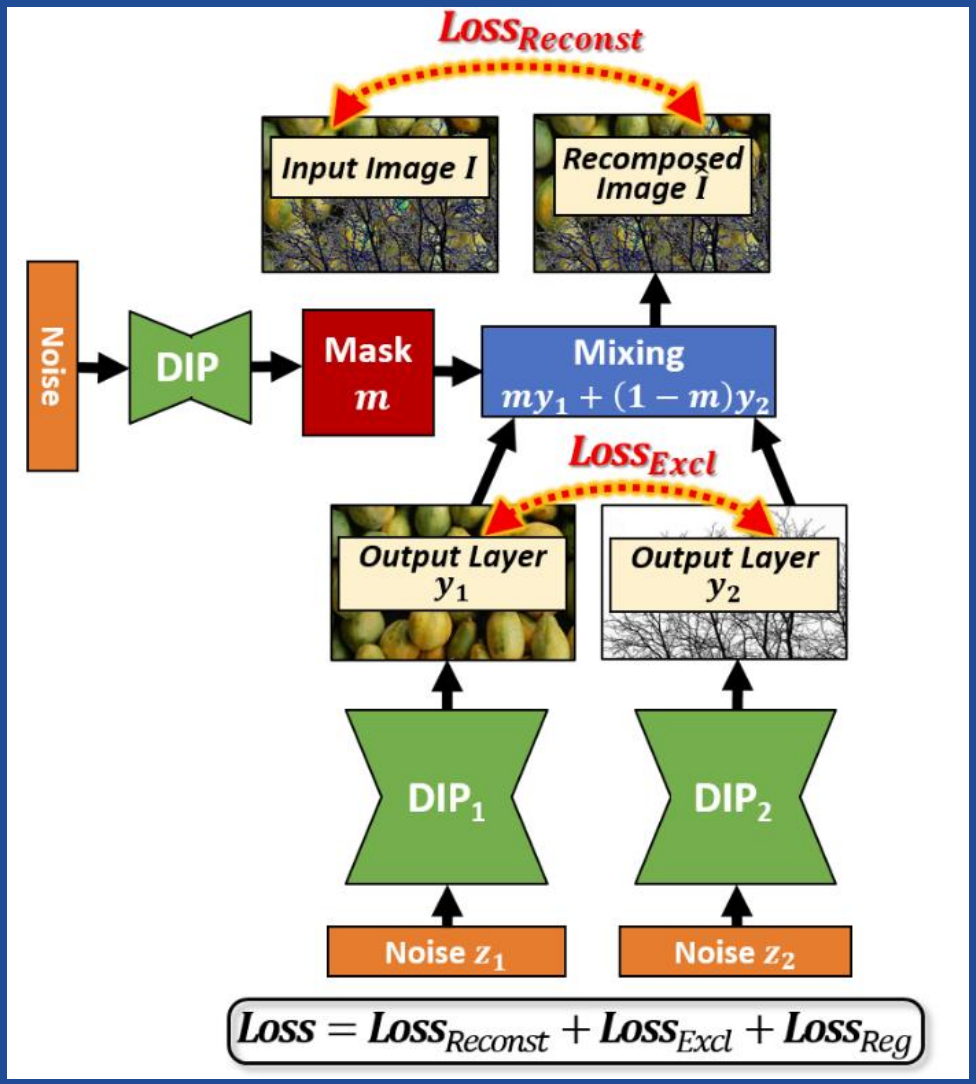
Computer Vision News

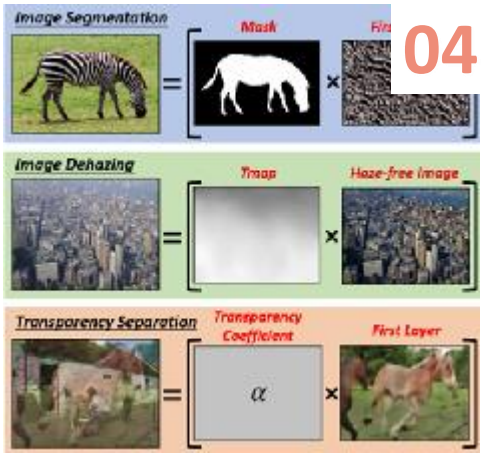
The magazine of the algorithm community

A publication by RSIP VISION

May 2019



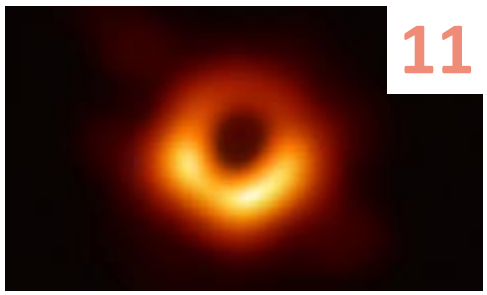
Contents



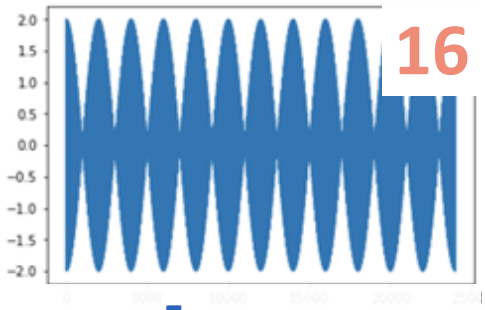
Research Paper Review
Double-DIP

Project by RSIP Vision
Airways Segmentation

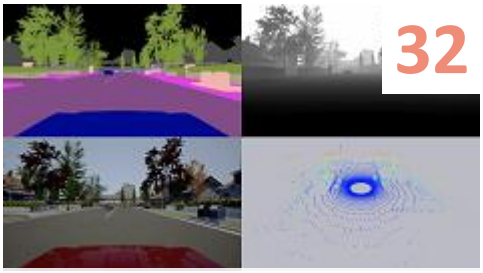
Women in Computer Vision
Gulcin Caner



AI Spotlight News



Focus on
JupyterLab



Challenge and Workshop
CARLA (CVPR 2019)

Project Management Tip
Crisis Management

We Tried for You:
Keras Pre-trained models

Upcoming Events

Did you subscribe to Computer Vision News? It's free, click here!

Computer Vision News

Editor:
Ralph Anzarouth

Engineering Editors:
Assaf Spanier
Amnon Geifman

Publisher:
RSIP Vision

Contact us

Give us feedback

Free subscription

Read previous magazines

Follow us:

Dear reader,

Again, a content-rich issue of **Computer Vision News** is offered to your reading. In this May issue, you will find a multitude of articles written by my colleagues at **RSIP Vision**. Some of them are more technical and more challenging to read: we publish them for the more advanced AI professionals. Other articles are fit for a larger audience of readers. You can shape the next issues with us: here is a form to give us your **feedback**. It will help us at RSIP Vision design our magazine according to your taste, as we have already done for **more than three years**.

The Bay Vision group, which we sponsor, keeps hosting monthly meeting in the different fields covered by **Computer Vision, Deep Learning and Artificial Intelligence**. If you work in the Bay Area or if you happen to travel there, join the group and come over. **The pizza is on us!**

Not earlier than yesterday, **Moshe Safran** and **Miki Haimovich** have hosted another webinar; this time it was dedicated to the topic of "How to boost your medical application with AI". Did you miss it? No worries, here is **the full recording (with all the slides)**.

Finally, please let me remind you to **share this magazine with colleagues and friends**. You can also subscribe them for free (see page 40).

Enjoy the reading!

Ralph Anzarouth
Editor, **Computer Vision News**
Marketing Manager, **RSIP Vision**

by Assaf Spanier

Every month, Computer Vision News reviews a research paper from our field. This month we have chosen **"Double-DIP": Unsupervised Image Decomposition via Coupled Deep-Image-Priors**. We are indebted to the authors (**Yossi Gandelsman, Assaf Shocher and Michal Irani**), for allowing us to use their images. The paper is found here.

Many apparently unrelated computer vision tasks can be thought of and dealt with as special cases of decomposition into separate layers.

To name just a couple of prominent examples:

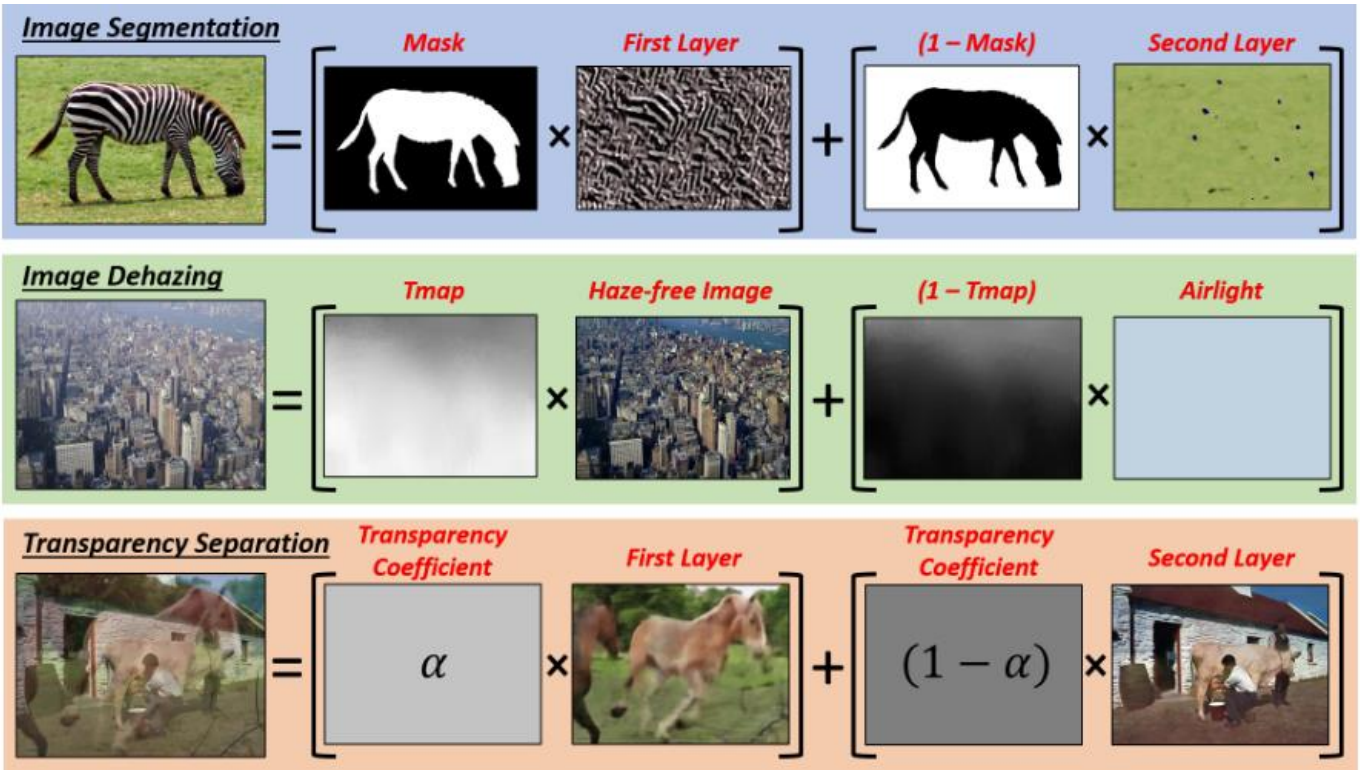
- Image segmentation -- which can be defined as decomposition into areas belonging to a background layer and areas belonging to a foreground layer.
- Image dehazing -- which can be defined as decomposition into a clear, clean image and the dehazing map layer.

The authors propose a unified framework for unsupervised layer decomposition of a single image, based on Deep-image-Prior (DIP) networks.

Deep-image-Prior (DIP) networks, introduced at CVPR 2018, are a type of generative network that learns the low level statistics of a single image -- is trained on a single image. In the article, the authors show how stringing together several DIP networks provides a powerful tool for decomposing images into their basic elements -- for a wide variety of tasks. The authors believe this versatile applicability derives from the fact the internal statistics of a mixture of layers is more robust and has better representation capabilities than each layer separately.

The authors show the applicability of their approach to a variety of computer vision tasks, including watermark-removal, Fg / Bg segmentation, image dehazing and transparency decomposition in video images, among others. All of these capabilities are achieved when the network is trained on a single image with no additional data provided.

A unified framework for image decomposition -- below are illustrations of the article's approach in action. Three different tasks redefined as decomposition of the original image viewed as a mixture of simpler basic layers. This approach of image decomposition into a number of basic layers -- provides a unified framework for dealing with a wide number of apparently disparate and unrelated computer vision tasks.

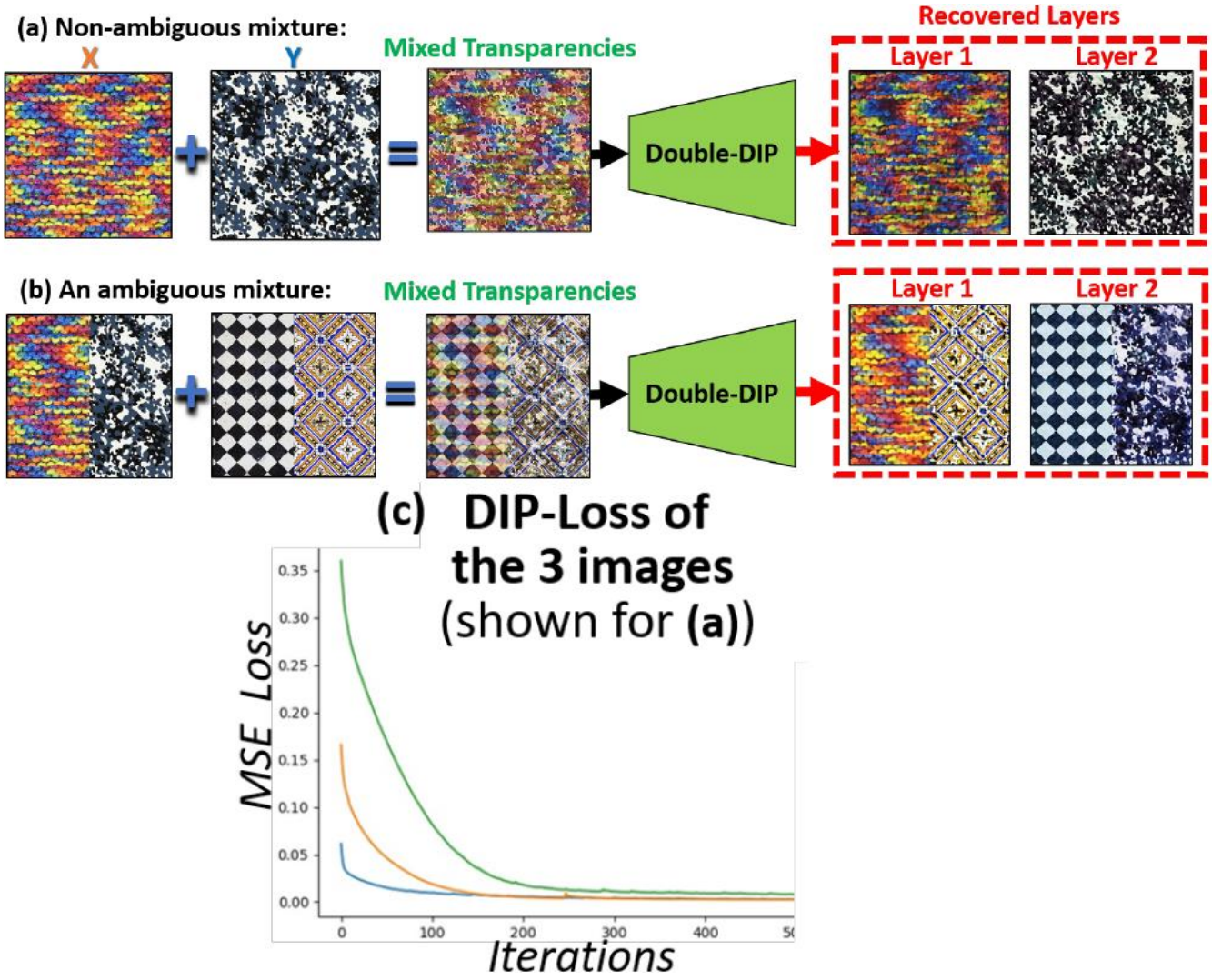


What all these image decompositions have in common is the fact that the distribution within small patches within each layer is "simpler" (more uniform) than for small patches of the "mixed" (original) image, resulting in a strong internal resemblance for each layer.

The statistical characteristics (distribution) of small image patches (such as 5x5, 7x7) have been proven to widely repeat in a natural image. This strong internal repetition has been exploited for tackling a wide variety of computer vision tasks.

The authors' approach combines the power of small image patches repeating throughout the image (its power of solving tasks without supervision) with the power of deep learning, and they propose a robust, unsupervised framework, based on DIP networks. A single DIP network was shown to be sufficient to capture the low level statistics of a single natural image, when the input for the DIP network is random noise, and the network learns to reconstruct a single image (the image serving as the sole input for training the network). This network proved to be powerful enough to solve problems such as denoising, super resolution, inpainting, doing all this without supervision.

… the power of small image patches repeating throughout the image with the power of deep learning

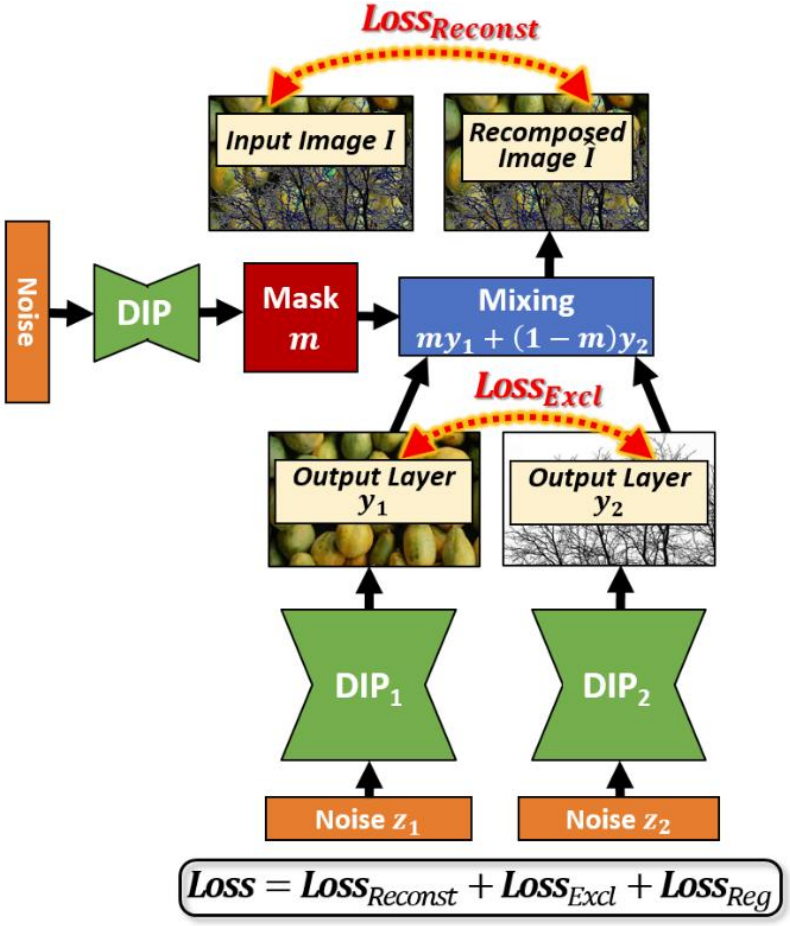


The figure above demonstrates the basic idea behind the approach -- it shows two patterns -- X and Y, mixed together to produce a more complex image -- Z. The distribution of small image patches for each “pure” pattern (X and Y) is simpler than the distribution of small image patches for the mixed image Z. We know from literature that if X and Y are two independent random variables, the entropy for their sum Z = X + Y is greater than each of their respective entropies.

The figure also shows a graph of the MSE **loss** for image reconstruction using a single DIP network as a function of time. Three image reconstructions are plotted: (i) the orange line represents the MSE loss, when the DIP network is trained to reconstruct image X; (ii) the blue line represents the loss when trained to reconstruct image Y; and (iii) the green line represents the loss when trained to reconstruct the mixed transparencies image X+Y. You can see that the higher the starting loss value, the longer it takes the network to converge. The loss of the mixed image is not only higher than the loss of each of its component images, it is in fact higher than the sum of their loss values. This is attributed to the fact that the distribution of small image patches in the mixed image is more complex and diverse (higher entropy, lower internal similarity) than in either component image.

This phenomenon proved to be consistent, so the authors repeated the experiment on 100 randomly selected pairs of natural images from the BSD100 dataset -- and in the case of natural images rather than patterns, there was an even higher disparity in MSE loss values between the mixed image and its component images.

Method:



The figure above demonstrates the Double-DIP Framework: two Deep-Image-Prior networks (DIP1 & DIP2) decompose the input image *I* into its layers (y1 & y2), then those layers are recomposed according to a learned mask m, reconstructing an image $\hat{I} \approx I$.

What constitutes a good image decomposition? There are many ways to decompose an image into basic layers, however, the authors propose the following characteristics as defining a meaningful decomposition:

1. The recovered layers, when recombined, reconstruct the input image.
2. Each layer should be as “simple” as possible, that is, it should have a strong internal self-similarity of image elements.
3. There should not be dependence between the recovered layers, or there should be the least correlation possible between them.

These criteria were used as the basis for a general-purpose Double-DIP architecture:

- The first criterion is enforced by minimizing Reconstruction Loss, which measures the error between the recomposed image and the input image.
- The second criterion is achieved by employing multiple DIPs - one per layer.
- The third criterion is maintained through the Exclusion Loss between the output of the different DIP networks minimizing their correlation.
- Each DIP network recovers its own layer y_i of the input image *I*. The input to all DIP_is is randomly sampled noise - designated z_i. The outputs of all the different DIP networks-designated $y_i = DIP_i(z_i)$ are mixed via a weight mask to create a recomposed image:

$$I = m \cdot y_1 \quad + (1 - m) \cdot y_2$$

  which should be as close as possible to the input image *I*.

For some tasks the mask *m* is simple and known, in other cases it must be learned (using another DIP network). The learned mask can be uniform or have varied characteristics for different areas of the image, can be continuous or binary. These constraints for *m* are task-dependent, and enforced through **tailoring** the Regularization Loss **term and parameter** to the task. Thus, the Optimization Loss is:

$$Loss = Loss_{Reconst} + \alpha \cdot Loss_{Excl} + \beta \cdot Loss_{Reg} * -0.1cm$$

where $Loss_{Reconst} = ||I - I'||$ and $Loss_{Excl}$ (the Exclusion Loss) minimizes the correlation between the gradients of y_1 and y_2. And $Loss_{Reg}$ is a task-specific mask regularization.

The training and optimization of a double-DIP network are similar to those of the basic DIP. The addition of extra noise to the input gradually increased with iterations, increases the robustness of recomposition, as does data augmentation of the input image *I* by the addition of noise produced by 8 transformations (4x 90° rotations and 2x horizontal and vertical reflections).

Optimization uses the adam optimization tools and takes a few minutes per image using a Tesla V100 GPU.

Results:

Of the many results described in the article -- we will discuss two:

1) image segmentation into Fg / Bg, and
2) watermark removal.

Image Segmentation:

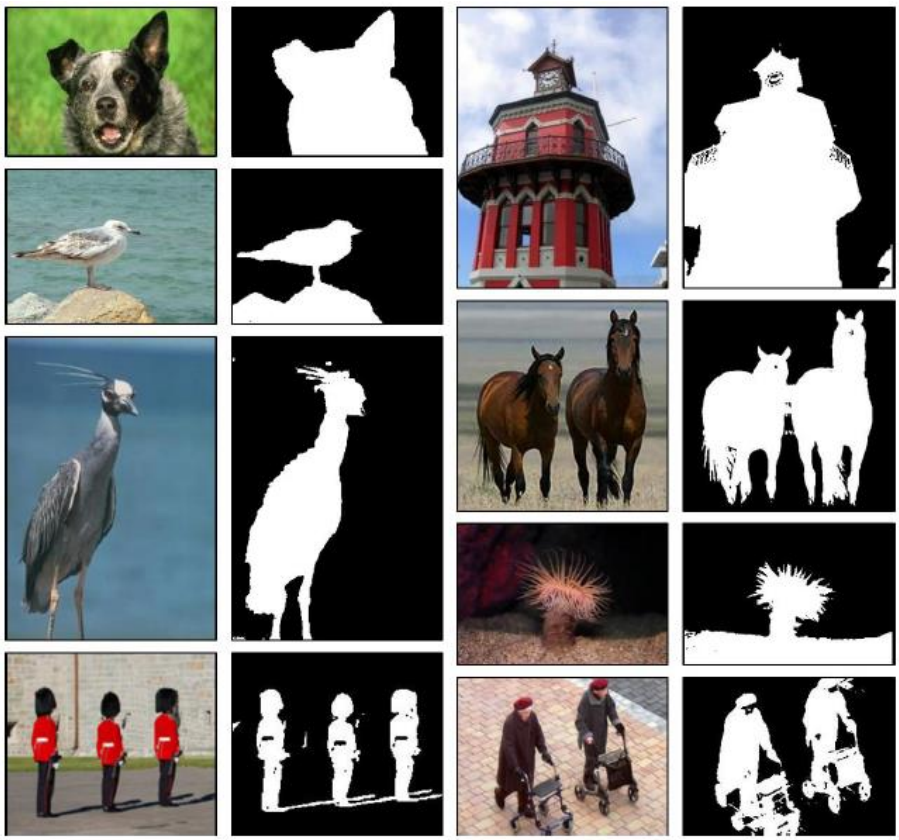
Image segmentation into foreground and background regions can be conceived of as decomposition of the image into a foreground layer y1 and background layer y2, combined according to a binary mask m, which for every pixel x follows the formula:

$$I(x) = m(x)y_1(x) + (1 - m(x))y_2(x)$$

This formulation naturally suits the framework proposed by the article, which defines a 'good image segment' as one that is easily put together using parts that belong to it, but hard to put together using parts from other segments of the image. To encourage the Segmentation Mask to be binary, we use the following loss term:

$$Loss_{Reg}(m) = (\sum_x |m(x) - 0.5|)^{-1}$$

Double-DIP can achieve high quality segmentation based solely on layer decomposition, with no supervision, as illustrated in the figure below:

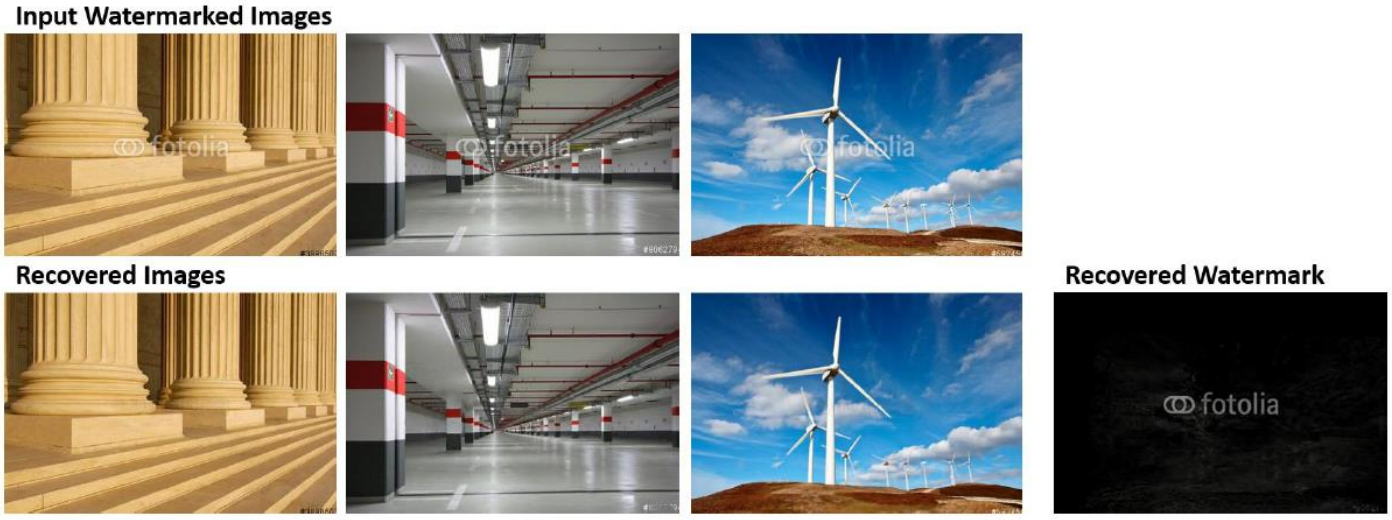


Many more results can be viewed on the project's website. It's true that there are many other approaches to segmentation (among them, semantic segmentation), that perform even better than DIP. However, they all suffer from the disadvantage that they need to be trained on large datasets.

Watermark Removal:

Watermarks are in widespread use for protecting copyrighted images and video footage. Double-DIP can remove watermarks by treating them as a special case of image reflection, where layers y1 and y2 are the cleaned-up image and the watermark, respectively. In this case, unlike for segmentation, the mask is not set, and inherent transparent layer ambiguity is dealt with by using one of two practical solutions:

1) If only a single watermark is involved -- the user provides a cue by marking the area with the watermark using a bounding box;
2) When there are a few images with the same watermark (2-3 images are usually enough), the ambiguity resolves itself in the training process. The decomposition of a few images sharing a watermark is illustrated below:



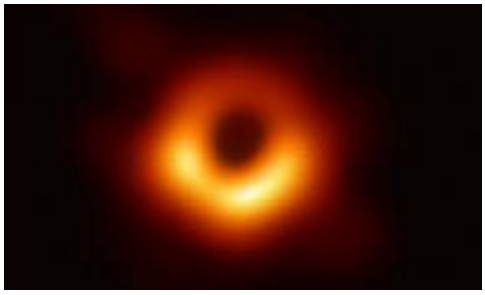
Conclusion:

Double-DIP provides a unified framework for unsupervised layer decomposition of a single image. The proposed framework is applicable across a wide variety of tasks. The framework doesn't require additional data beyond an input image / video. Despite the methods generality, on some tasks (such as dehazing) it achieves results equal or even better than current state of the art specialized methods in the field. The authors believe that augmenting Double-DIP with semantic/perceptual cues may lead to improved performance on semantic segmentation and other high-level tasks in computer vision. They intend to pursue this in future research.

...augmenting Double-DIP with semantic/perceptual cues may lead to improved performance on semantic segmentation and other high-level tasks in computer vision.

Computer Vision News has found great new stories, written somewhere else by somebody else. We share them with you, adding a short comment. **Enjoy!**

First Ever Observed Black Hole:

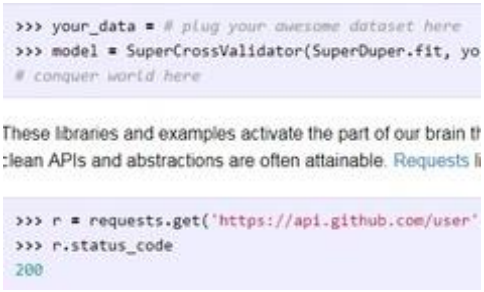


As you certainly know, the scientific world was caught by surprise a few weeks ago with the revelation of the first picture of a black hole. Those who were not surprised are the early readers of this magazine, since they read the story of this research almost 3 years ago, when we interviewed Katie Bouman of MIT at CVPR2016. **Read It...**

Open Questions about Generative Adversarial Networks:

Even those who think that **GANs are great** (and they are!) will enjoy this great article about Trade-Offs between *GANs and other Generative Models* and much more. **A Must Read!**

A Recipe for Training Neural Networks by A. Karpathy:



Many of people have personally encountered the large gap between "*here is how a convolutional layer works*" and "*our convnet achieves state of the art results*". **Andrej Karpathy** wrote a fascinating blog post to dig deeper into this topic and to give his personal recommendation to **networks trainers**. **Read More...**

DeepMind and Google: - the battle to control AI:

If you think you know a lot about **DeepMind**, its history, its management and its mission, prepare to change your mind. We learned A LOT from this article and we promise that you will learn too. **Read It!**

Google launches an end-to-end AI platform:

More about **Google**: the Mountain View company keeps expanding its AI services with the purpose of **democratizing AI and machine learning** with pre-built models and easier to use services, while also giving more advanced developers the tools to build their own custom models: an end-to-end **AI Platform** for building, testing and deploying their models. **Read More...**

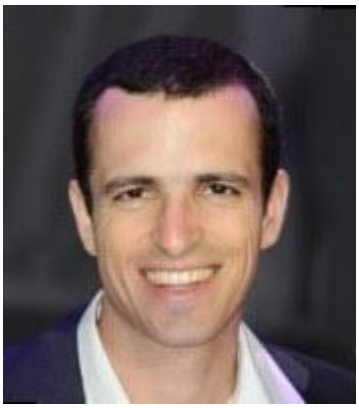
MorphNet: Towards Faster and Smaller Neural Networks:

Another must read post on **Google blog (by Andrew Poon** and **Dhyanesh Narayanan**)

Tree sleuths are using DNA tests and machine vision to crack timber crimes:

Great article by Nature on **innovative AI techniques** that help fight **illegal wood trade.**

Crisis Management



RSIP Vision's CEO Ron Soferman has launched a series of lectures to provide a robust yet simple overview of how to ensure that computer vision projects respect goals, budget and deadlines. This month **Nissim Avitan** tells us about another aspect of successful **Project Management: crisis management**. It's another handy tip by **RSIP Vision** for Project Management in Computer Vision.

When the development team delivers work to the client and/or to the testing team, it may happen that the software displays unwanted behavior or failure. We are talking about real-time software, a situation in which future input is unknown at the coding time. The development team should always keep in mind the possibility of a future crisis, which might occur later on.

Sometimes, developers have the tendency to work on the urgent task, like delivery, functionality, communication with the customer and so on. This propension could be magnified when the software is performing fine with current data: all the cases are working OK and the customer (or the testing team) is very much satisfied with the performance of the software. When future input is unknown, the project manager should always be prepared to the possibility of a crisis when additional input data is provided.

"...never go unprepared..."

A good solution to never go unprepared is the accuracy measure: a measure which could define how large is the margin which keeps our algorithm on the safe side, with regards to future input data. These accuracy measures will give information beyond the current performance with current data: they will tell how close the software is to its maximal capability; in other words, how easy it is for the software to provide the current performance and how large are the margins until it might meet a crisis.

A practical example for the accuracy measure: let's say that we have an image and we are looking for a circle inside it. We want to measure how close is the found circle to our model of circle. Do all the points in this circle fit with the given formula of a circle?

Once we have defined this accuracy measure, we need find a bulk of cases upon which test it and find out where we stand. We can add random changes to that bulk of results, so to include in the test also some levels of noise, that we expect to receive later in the real data. We increase the data complexity to check how far the software can still go.

Ideally, the project manager should never find the software too close to the limits: there should always be a double margin with respect to the current results. This work on the stability of the software must be done, regardless of the high pressure being

put upon the team for fast delivery and the like. This will help avoid situations of crisis further down the line. When the measure is close to the limit, the client should be informed and the work should be slowed down to enable the needed changes to the software.

In order to do so, the software architecture should be built in such a way to perform a measurable test of the quality of the results. Failing to do so might be comparable to building a very high tower, breaking several milestones. When the crisis will reach the software, it will be much deeper: it will then be much more expensive to correct, regardless of the software performance until then. Our recommendation is to dedicate part of the development time (let's say about 10%) to building a completely different approach. That means preparing in advance an approach which now might seem useless, too experimental or even too dangerous, but it might prove itself to be a valid alternative at a later stage.

More articles on Project Management

Challenges in Airways Segmentation

Computer Vision News

by Ilya Kovler

Every month, **Computer Vision News** reviews a successful project. Our main purpose is to show how diverse image processing techniques contribute to solving technical challenges and real world constraints. This month we review challenges and solutions in a Medical Imaging project by **RSIP Vision**: **Airways Segmentation with Deep Learning**.

Project

Airways Segmentation is a very useful development, which helps physicians in a multitude of tasks: diagnosis, operation planning, assessment of treatment results and more. It also presents several challenges that need to be solved, before the project is able to provide its benefit.

Some of this challenges are similar those found during other body parts segmentation. One of these is the wide variations of the anatomy between one patient and another: even though they are composed of the same parts, the details of these parts differ in many ways and this difference is mirrored in the CT scans.

CT scans come in very diverse resolutions and quality, depending on the machine involved and its producer.

RSIP Vision has a very long experience in solving these challenges. We have built a unique data augmentation system, able to generate additional CT scans, which look like real even when the anatomy has been deeply modified

The level of noise might change from one vendor to another and the algorithm needs to provide an accurate segmentation also when dealing with many of them.

Most often, CT scans are done on unhealthy people, whose anatomy is different. For instance, in a patient who has had surgery before, a metallic implant might cause artifacts in the CT.

Finally, the running time is a challenge too: the analysis of the medical scan needs to be fast. Some times it can run in a few seconds and in other cases several minutes are needed.

RSIP Vision has a very long experience in solving these challenges. We have built a unique data augmentation system, able to generate additional CT scans, which look like real even when the anatomy has been deeply modified.

Accurate annotation requires a huge effort. Even though we use partially automated tools, it takes radiology experts years of man work to annotate and validate a sufficient quantity of data.

RSIP Vision has a long experience in solving these and other challenges. We always use statistical solutions, based on machine learning and deep learning, rather than heuristic methods: the latter do not offer a proper solution when a the dataset includes very diverse cases. Instead we train and validate our solutions on a varied and balanced dataset, including thousands

of annotated CT scans. It is also important to include all kind of pathologies. We have developed a proprietary medical data augmentation system: modified CT scans look like real ones, even though the shown anatomy is very different.

Validated annotation requires a long work, even when we use semi-automated tools. The solution must be validated with the highest accuracy and objectivity. There must be a test set which does not participate in the training and which is never used by the R&D engineer to build the algorithm. This test set must be varied and large enough to be relevant and credible, at least as large as the training set.

Read about more projects in medical segmentation

Take us along for your next Deep Learning project! Request a call here

Project

by Assaf Spanier

This is Assaf's last article for Computer Vision News, on which he published almost 100. He now leaves the magazine to join a very different project. We owe to him much of what we achieved and we wish him a lot of success!

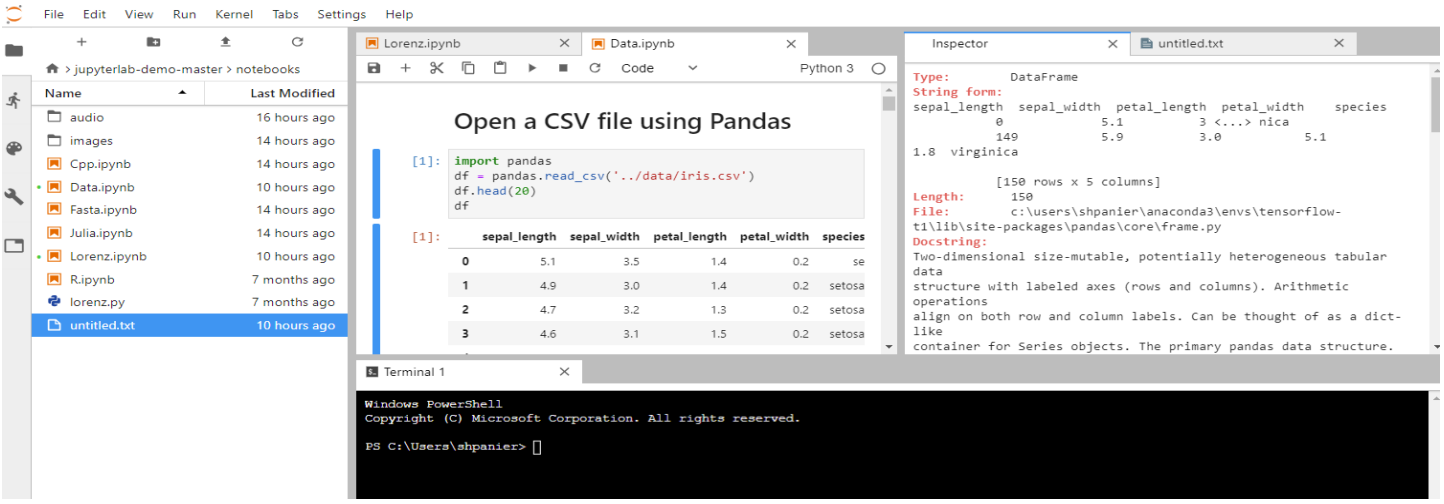
Jupyter is a tool for doing interactive computing with human-in-the-loop - something we tend to think of as not very useful and only half-respectable. However, it turns out to be essential for writing some code, inspect the results, and maybe get some rich outputs - in all of our fields: computer science, data science, and even for training networks in computer vision. The Jupyter project started in 2012 with the release of the ipython notebook. JupyterLab is the next generation of this project; it includes all features of the classic Jupyter Notebook with the addition of a flexible, powerful web-based interface which includes: notebook, terminal, text editor, file browser, rich outputs, and much more. It enjoys widespread use, with over 3 million notebooks publicly available on github and a roughly similar number of private notebooks - so this is a tool that has become very popular quickly.

Just to give you a quick taste of what it looks like, here is a first print screen from JupyterLab. The first big difference you'll notice is its multiple window approach, which gives it the look and feel of a "real" IDE. On the left you can see a file browser with a list of files, a list of running notebook sessions and a palette of executable commands. The middle panel should feel familiar to Jupyter ipython notebook users, with a lot of the same features, however, now supporting multiple tabs - multiple notebooks open at the same time. On the right - is the inspector - which you can use as a debugger which lets you inspect the types and values of your variables. On the bottom you can see a bash terminal which lets you run commands from inside the browser. There is also a text editor available.

Handy, isn't it?

Installation

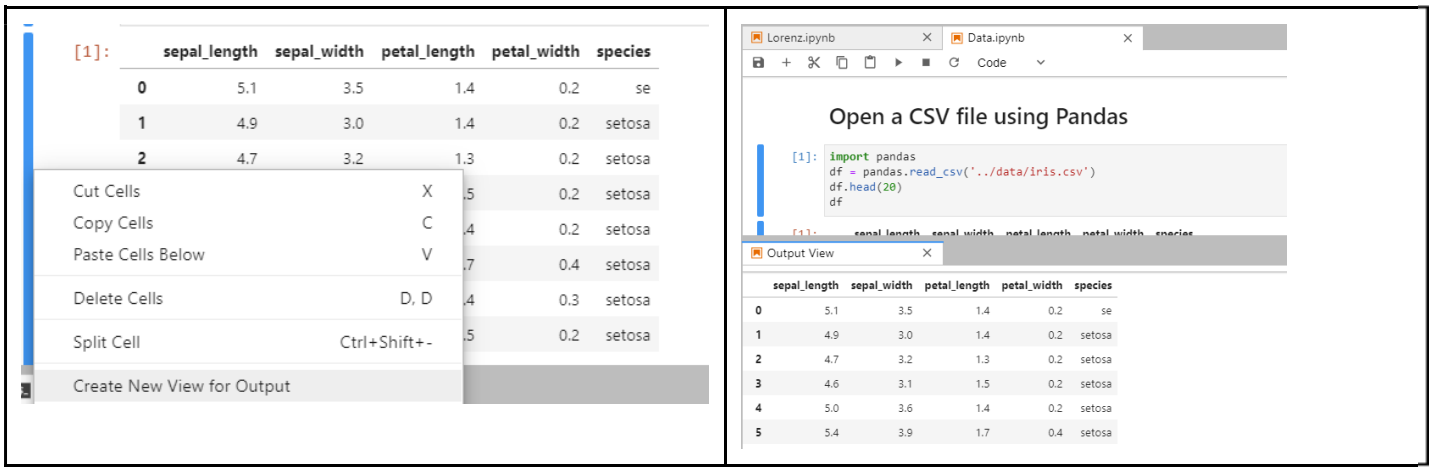
To set this up on your machine, follow the instructions on the next page:



1. Conda conda install -c conda-forge jupyterlab
2. Conda pip install jupyterlab
3. conda install ipywidgets
4. jupyter labextension install @jupyter-widgets/jupyterlab-manager
5. Then run it with “jupyter lab”

Ability to create a new view for the output:

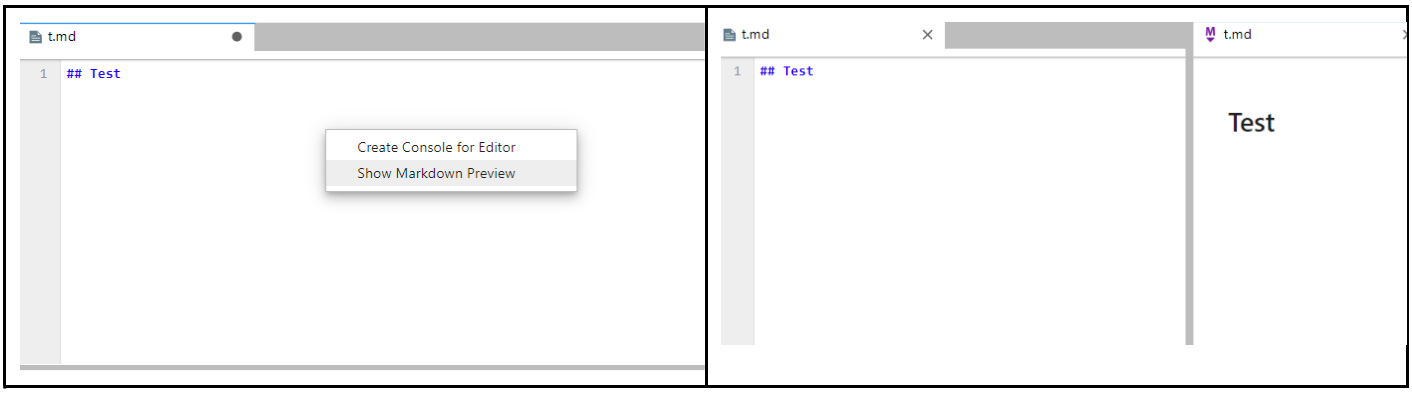
Having to scroll up and down to view the code and output can be really annoying, so it’s nice that one of JupyterLab’s features is the ability to create a new view for the output in a separate tab, so you can see things side by side. See below, the left illustration: right click to open a pop-up menu and select “Create New View for Output”, the right illustration: a code panel and an output panel. There's also dragging and dropping of cells within and between notebooks - try it out - it’s easy.



Focus on

Interactive markdown

To create an Interactive markdown, select “file”, then “new” and create a new text file - rename it to t.md to make a markdown file, add some text into it, for example “## Test”, as we did here. Now, right click and select “show markdown preview” (see left illustration below) and a markdown preview of the content of your file will pop up on the left (see right illustration below).



Running .py files

You can create and run .py files in JupyterLab, not just notebooks. To do this, create a text file, rename it to have a .py extension. Right click to open a Python 3 console dedicated to this .py file. You will be able to press shift enter in this Python code block and it'll execute and show you the outputs in the console. See illustration below.

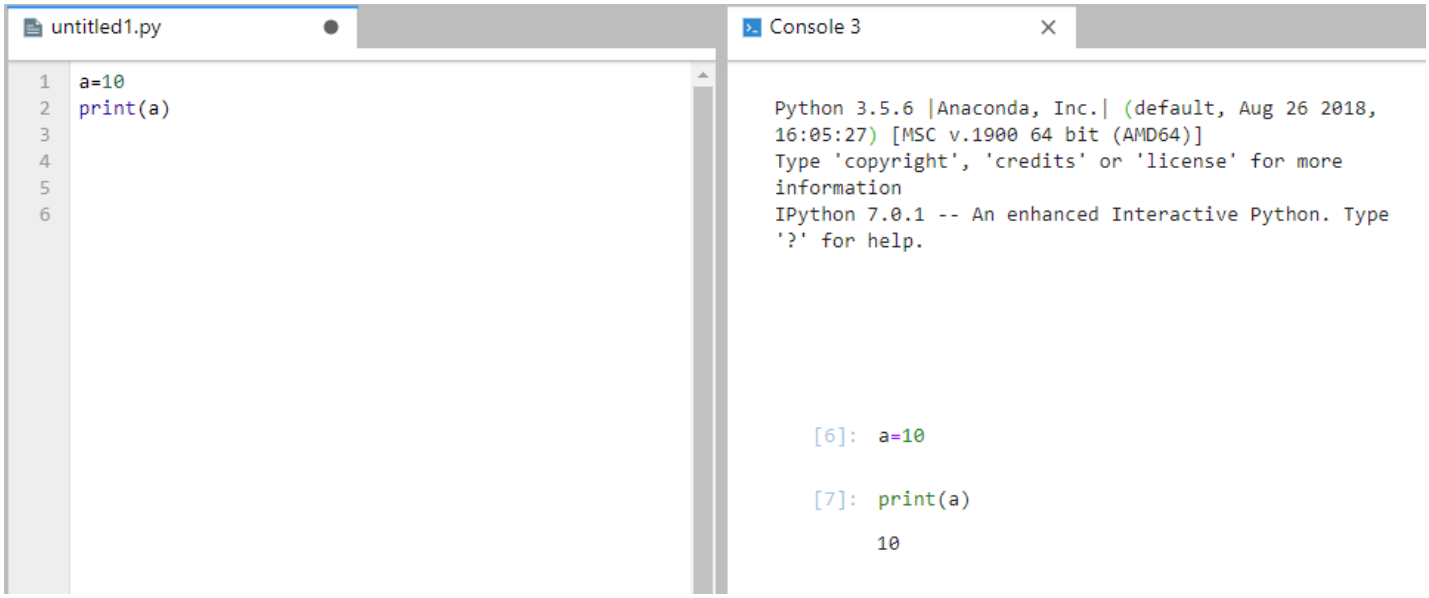
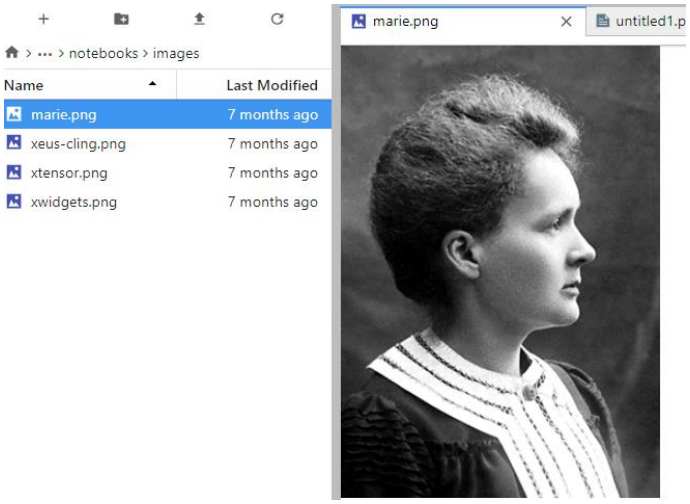


Image viewer

You also have an image editor - click on the image and a preview will pop up. As you can see, the idea is that you will have all the tools that you need to be available without the need to leave the JupyterLab web-based interface, even down to something like an image viewer.



What else? You get cool interactive widgets which enable you to scroll to change the parameters of your function on the fly and see the results:

```
from ipywidgets import interact

def my_function(x):
    return x

# create a slider
ww=interact(my_function, x=10)
display(ww)
```

x 6

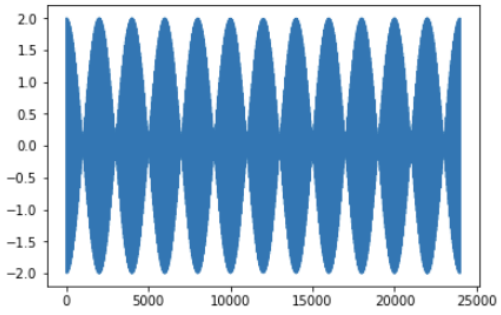
```
[4]: import matplotlib.pyplot as plt
import numpy as np
from ipywidgets import interactive
from IPython.display import Audio, display
def beat_freq(f1=220.0, f2=224.0):
    max_time = 3
    rate = 8000
    times = np.linspace(0,max_time,rate*max_time)
    signal = np.sin(2*np.pi*f1*times) + np.sin(2*np.pi*f2*times)
    print(f1, f2, abs(f1-f2))
    display(Audio(data=signal, rate=rate))
    try:
        plt.plot(signal); #plt.plot(v.result);
    except:
        pass
    return signal
v = interactive(beat_freq, f1=(200.0,300.0), f2=(200.0,300.0))
display(v)
```

f1 220.00

f2 224.00

220.0 224.0 4.0

0:01 / 0:03



There is an entire community of developers creating more and more optional extensions for you to install. This option for add-ons to be created by the developer community is a new one for JupyterLab. The following is a short list of the most popular extensions and how to install them (if you need them):

Focus on

1. Vega and Vega lite documents is a high-level grammar for interactive graphics.
2. High-performance csv viewer one of the cool things about this csv viewer is its extremely high performance, it can render large CSVs because it does lazy loading and lazy rendering.
3. Built in github viewer for JupyterLab so you can use your github username to log on to github and upload your data directly into your JupyterLab session.
4. **JupyterLab LaTeX -** allows you to view and edit LaTeX files inside JupyterLab. Install this extension with the following commands:

```
# to install server extension
pip install jupyterlab_latex
# to install jupyter extension
jupyter labextension install @jupyterlab/latex
```

   Then, all you need to do is create a text file, change its extension to .tex, then right click and select Show LaTeX Preview.

5. **JupyterLab HTML -** Jupyter allows you to use HTML files, simply click on the html file to open it. Install this extension with the following commands:

```
For installation type:
jupyter labextension install @mflevine/jupyterlab_html
```

6. Clean your code with one click. This extension automatically formats Python code to conform to the PEP 8 style guide, allows you to simply click the gavel and automatically format your code.

```
For installation type:
jupyter labextension install @ryantam626/jupyterlab_code_formatter
```

7. **Variable inspector -** keep track of your workspace. The variable inspector enables you to collect all defined variables and display them in a floating window. It shows all variables you've created in the notebook, along with their type, size, shape, and value.

```
jupyter labextension install Variable-inspector
```

8. **ExecuteTime -** shows when and how long cells ran. ExecuteTime displays the time at which the user sent the cell for execution, and how long it took to finish.

```
jupyter labextension install upyter-contrib-nbextensions.readthedocs.io.
```

RSIP VISION

Global Leader in Computer Vision & Deep Learning

COMPUTER VISION PROJECT MANAGEMENT

Computer Vision Project Management is a series of lectures and articles conducted by RSIP Vision's CEO Ron Soferman, many of which are published as a regular column on magazine Computer Vision News, in the project management section.

Everything a project manager in computer vision should know... **at the click of a button**

- How to implement Deep Learning
- Team Leadership and Management
- Validation and Test Techniques
- How to solve all kinds of challenges
- What are the best practices?

"Even the biggest hammer cannot replace a screwdriver!"

Did you miss an article? No worries, you can find them all in the **Project Management section of RSIP Vision's website**

by Amnon Geifman

Image Classification Using Keras Pre-trained Models

Image classification is a basic task in computer vision and in many cases, using a pre-trained model can help us avoid the hard work of training and evaluating our models. Luckily, numerous pre-trained models are available online. In this article, we will review the Keras applications; deep learning models available on Keras, alongside with pre-trained weights. We will start with a quick introduction to Keras and its deep learning models for image classification, then implement a ResNet50 network on Keras applications, and finally demonstrate the performance of the model by classifying some real-world images.

Introduction

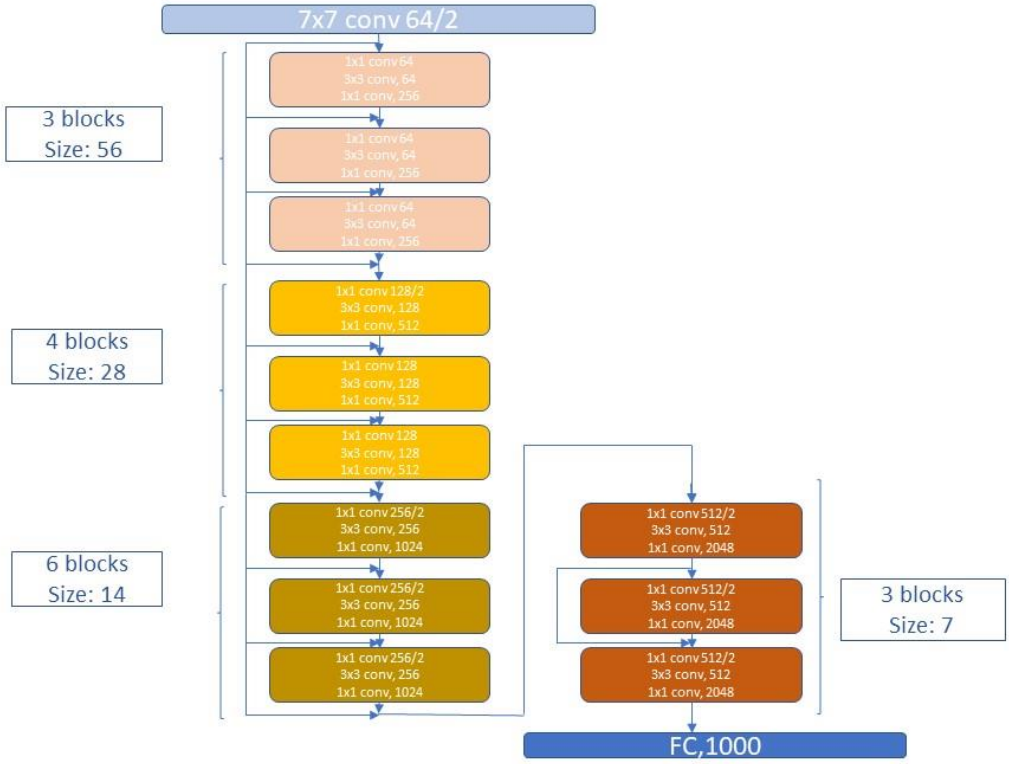
Keras is a Python library, that provides a high-level deep learning API that works on top of existing libraries such as TensorFlow, CNTK, and Theano. In practice, this means that instead of writing lengthy, low-level programs with any of these libraries, we can implement our model in a much more concise way. For example, training a basic model on TensorFlow requires us to define the loss manually, then the optimizer with its parameters, then split the data to batches and then iterate through the batches and epochs to train the network. Any additional configurations need to be defined separately. With Keras however, training a model after setting the structure of the model object can been done by two lines of code:

Tool

```
model.compile(loss='mean_squared_error',
        optimizer=optimizers.SGD(lr=0.01, decay=1e-6, momentum=0.9),
        metrics=['accuracy'])
model.fit(X, Y, epochs=5, batch_size=32, validation_split=0.3)
```

where X, Y are the features and the target respectively.

Another advantage of Keras is the availability of widely-used architectures as part of the API. Within Keras application, one can find image classification architectures such as: Xception, VGG16, VGG19, ResNet, Inception, MobileNet, DenseNet and more. This article will demonstrate the popular ResNet50 application and the power of skip connection. The architecture of ResNet can be seen in the following figure:



The ResNet architecture:

Implementing ResNet50 in Keras

A common issue when using a pre-trained model is to adjust the data to the desired input of the network. For example, in the ResNet50 architecture, the input image is a 224x224 image. Moreover, the Keras model requires the input of the model to be of a specific size (dimensions) and normalized. Fortunately, we do not need to implement such preprocessing by ourselves, we can use pre-defined utility functions that are available in keras.application, numpy and keras.preprocessing. We first import them to our environment:

```
from keras.preprocessing import image
from keras.applications.resnet50 import preprocess_input, decode_predictions
import numpy as np
```

Now that we have the image class in our environment, we can load the image from the disk, and with the same command perform resizing of the image to the size of 224x224. Using the image subclass of Keras it looks like this:

```
img_path = 'ImagePath/ImageName.jpg'
img = image.load_img(img_path, target_size=(224, 224) )
```

Focus on

The variable resulting img on the second line is a Keras image object of size (224,224).

Next, we would like to change the dimension of the image. To this end, we will add one channel to gray images and three channels to RGB images. Since the network gets as input batches, we will also create a single image batch from our input image. These two operations can be done efficiently by using these two lines (respectively):

```
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0 )
```

Finally, we will normalize our input. As we all know, input normalization is essential to training a deep network. Normalization can take different forms: some to a range of 0 to 1, some to a range of -1 to 1, some subtract the mean and divide by the variance and some are just centering by subtracting the mean. Here, ResNet50 requires that we will subtract the mean of ImageNet, so we use the following utility function of the form:

```
x = preprocess_input(x)
```


And now we are ready to start using the model. We will use the following command to import the ResNet50 model and to define our ResNet50 model object :

```
from keras.applications.resnet50 import ResNet50
model = ResNet50(weights='imagenet')
```

In our case we chose the pre-trained weights trained on ImageNet; however, users who want to use the architecture and train the model themselves can set the 'weights' argument to None. This, will provide a random initialization to the model. For example, we would use this option if we wanted to train a model on different data set. Note that it is also possible to initialize the weights using the ImageNet pre-trained weights, and using this initialization to further optimize the weights for a different task.

Finally, we produce a prediction from our model by simply writing:

```
preds = model.predict(x)
finalPrediction = decode_predictions(preds, top=3)[0]
```

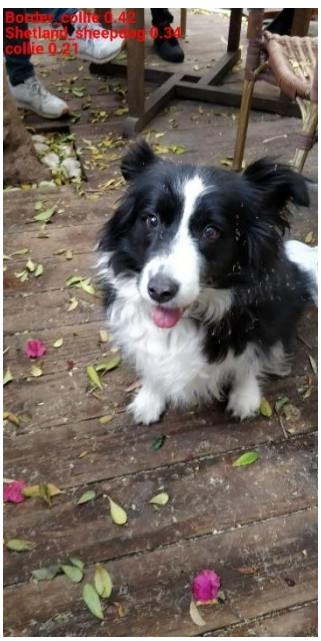
In this article, we chose to use the top three classes. Choosing the parameter 'top' to be 3, means that we take the multiclass prediction of the network, which is a number associated with each class, use the three argmax of this vector. Those will be the three classes of our prediction where each class gets a score for the input image being in this class. We also implemented a simple code that by using pillow library, shows the image, write the prediction on the image and save it to the disk.

At this point, we have the model in our hands, and we can use it to classify images. Let's see some results: Dog breed test usually requires a blood test, which might cost a significant amount of money. In this example we will show how to make it much cheaper.

ImageNet dataset contains about 1000 classes some of the classes are objects, some are plants, some are animals, and some are animal species and subspecies. In this example, we will classify dogs into breeds by using the 120 classes of different breeds of dog that are available in ImageNet. The output of the network is a vector with 1000 entries, each of the entries corresponding to a specific class. Our prediction for the dog's breed will be the 3 highest ranking classes.

Below are some results of our model. The network top 3 predictions and the score for each class are listed at the top left of each image. One can appreciate the accuracy by examining our visual results. Obviously, for more accurate classification we will need to develop a much more complex model; however this is a good start!

This is a good start!



Gulcin Caner

Gulcin Caner is senior computer vision scientist at Altia Systems in Cupertino, California.

[Read more interviews with women scientists]

Gulcin, can you start by telling us about your work?

Yes, sure. Here, I am working as a computer vision scientist, and I am responsible for designing new algorithms to teach our multi-sensors. We have three sensor and six sensor cameras, and last year we developed this virtual reality camera which has six sensors for Intel. I had the opportunity to go to NBA games and use the camera for live streaming. It was lots of fun to see people watching it from their homes. The pictures captured by my camera were amazing.

You said "here". But our readers don't know where "here" is. Where are we now?

[*we both laugh*] Sure, of course! Here is Cupertino, California, and I've been working here for three years.

And the company is? Let's not hide it!
No, no! Altia Systems!

How did you get here, and why are you

doing what you're doing?

I have a PhD in electrical and computer engineering from the University of Rochester. In 2006, I got my degree. Then I moved to California from Rochester to pursue a research scientist job at another startup called Polar Rain. I worked for Polar Rain for exactly ten years. I flew up to Boston twice a year to the Raytheon Company, one of the biggest defense companies in the US. I collaborated with a number of universities: the University of Southern Florida, the University of Maryland, the University of Buffalo. It was great to be able to do both academic and product work. Then I started to work at this new company, Altia Systems. They found me, somehow, from LinkedIn or somewhere. I thought they were a nice option for me to continue my career. Here, it is very different from my previous job where I was responsible for software development. Here, I am responsible for hardware as well, in addition to software. I had to understand some hardware, firmware, and FPGA work. Now there's some AI work that started here about a year ago. The goal is to make our cameras smarter by letting them recognize people and names. I'm looking forward to learning more in the AI area as well.

Should we tell readers the great news about Altia Systems? What happened a few weeks ago?

[*laughs*] It started last year, actually, and it was confidential. In December, we had a few companies who were competing to buy us. This company from Denmark called GN Audio offered the highest price: six times the annual profit of our company, something never happened before in the Silicon Valley. They gave a very high evaluation. They would like to open a new center here and get into the video business.

This is not the first time that you have such an experience. Is this already your third time?

[*laughs*] Not the third time, no, that's exaggerating. In my previous company, my husband was working with my boss and started a company called Anvato in 2008. They have been acquired by Google about 2 years ago. Since he is also one of the founders, it was a very big exit for us financially. I mainly worked with family-based companies.

People are really hard working here, and they are extremely kind and very humble. We always have time to do the real work and not worry about the little things, who is getting ahead and so on...

You are not a real Californian, and you were not born in Rochester [*laughs*]. Where do you come from?

Yes, true. I am from Turkey. I was born in a town called Elazığ, on the east side of Turkey, and I studied in Ankara at the Middle East Technical University, which is the second top university in Turkey. I got my Bachelor's Degree in electrical engineering in 2000, and since 2000 I've been living in the United States. It's been a long time.

Did you want to be an engineer? Was it your dream as a kid?

I was very interested in math and science. Engineering was my first option. I was actually always a top student in my schools.

What made you the top student?

Yeah, everything was easy for me. I started school younger than my classmates. I was five, and I was reading in elementary school. I loved teaching, even in elementary school. Teachers were letting me teach stuff to my friends in their stead, even through middle school and high school.

You choose a career in the industry. For that, you had to quit academia. Was it a difficult choice?

No, I wanted to do it because, during my PhD, my advisor was not with me so I had to do my PhD almost alone. He was from Turkey, and he decided to go back to Turkey to work there. I had chosen core advisors in Rochester who could help me do my dissertation, and I was frustrated because I didn't have much direction. I didn't want to go forward in academia after I got my degree. If I had the option now, I might undo my postdoc and see if academia was right for me or not. I love the teaching side. I always want to go back.

Do you miss teaching?

Yes, yes. I teach my kids stuff.

Do you want to make them engineers?

[*laughs*] No, I just want them to be happy. I don't want them to worry about competition.

What is the main lesson in life that you are teaching them?

I want them to find things that they like to do. Whatever it is. So, I try to be an example of our learning-based stuff; you know, kids learn from us. So, if I ski, they try skiing. If I play piano, they do that. They think I work a lot so I think they will also work a lot, but I want them to enjoy friendships as well as academia.

What do you miss the most of Turkey?

[*laughs*] The desserts! Baklava! Yes! That's right, baklava [*both laugh*]

I don't like baklava.

Don't you like sweeties?

I don't. My father loves baklava.

I also miss my friends and whenever I go to Turkey, I go to my university and

meet with my professors. After 19 years, I still go visit them. Some of them became department chairs. Some of them retired, and I actually collaborate with them on some projects. I bring their students here. I always loved my teachers at university and high school, back home.

Can you see yourself living again in Turkey?

Unfortunately not. But I miss Turkey and I go there every summer for the Mediterranean Sea. [*laughs*]

So, you are a Mediterranean girl?

Yes [*laughs again*].

And I am a Mediterranean boy. We have something in common. What is the best thing that you have found here in America?

It's going to be cliché maybe, but people are encouraged to do learning at any age, which is not our eastern culture. It doesn't matter how old they are, or it doesn't matter which career they are in, they are encouraged to choose a different area or to keep learning. Also for innovation, you can come up with different ideas because people here don't think you're crazy to think out the very different opinions on a product. So, I think that is the best thing.

There is something interesting that you keep saying. You talk about yourself saying, "Now I am learning to do also this." You talk about your kids saying, "They are learning this and this." Then I ask you about America, and you say, "Here, people learn at any age". Is learning so important in your life? Is it so central to you?

[*laughs*] Oh, you are a psychologist! Yes, it is true actually. When I was doing my PhD, I used to tell my friends that studying is the best thing that anybody can do in this life. Some people like to travel in their life. Some people like to eat for their life…

The title of this interview will be: "*Studying is the best thing that anybody can do in this life*".

[*both laugh*]

You live for learning.

Yes, I live for learning.

What is the one thing that you have not learned yet that you would dream of learning, possible or impossible?

I would love to play an instrument, which I cannot play. I would love to play piano.

Why can't you?

I took classes when I was a child, and I stopped. Now, I am turning 40 this year, so I want to start again. Also painting, I

want to start painting.

Did you ever paint?

When I was a child, mostly oil and gouache paintings.

Did you keep them?

They're in Turkey, yeah. My mom keeps them.

Let's go back to work. From all your years working in industry, what did you learn that you would have preferred to have learned before? Perhaps something that academia did not teach you about life in the industry as an engineer that you think you should have learned before and you didn't.

When I was fresh out of my PhD, I had this very stubborn opinion of trying to model everything in the mathematical sense such as during my PhD in image processing. We were modeling what's happening inside of the camera with different equations, and when I started my job here in California after PhD, my boss told me: "*No, we don't do that!*" So, we are going to come up with an example-based learning. At the time, in 2006, AI wasn't this strong, and a professor from MIT published a paper about example-based learning. That paper started many things in our company. Then I thought of how I only have this one way of looking at problems, while there are so many different ways of looking.

So, the many years at school were not enough to give you the panoramic sight that you needed to succeed in industry?

Yeah.

Students will read this interview: what should they do in academia? What advice can you give them for succeeding in industry?

First, I tell them not to get stuck in their solutions or in the problems. I would advise them to see industry-based solutions, if they can, not just papers and presentations. Maybe go to industry shows and ask questions about the products because most of the papers that they see at the conferences never become products in the real world.

Very often, companies do not have enough time for students at trade shows, because they do not go there for them. It's difficult for a student to go to a show and say *"Please help me!"* They have teachers for that!

That's true. It is a very good idea to keep an open mind because PhD is a very frustrating time, and most people get stuck. They think that they cannot solve the problems or bring a new idea. Ideas come from many places so they should start every day with a new hope that this thing will happen today.

Jessica Sieren told me in her interview that if you don't have a crisis during your PhD, you are not doing a PhD. Do you agree with that? [*both laugh*]

Yeah, my mom just told me at lunch: "*Do you remember, Gulcin, you wanted to go back to Turkey during your PhD?*"

Is that true?

Yeah. So, I had a crisis. [*laughs*]

However, you got your degree. How did you solve the crisis?

It was in 2003, 3 years into my program, and I was very frustrated or upset, so I went to Turkey all of a sudden. My parents were very, very surprised. They were like: "*What are you doing here? It's not Christmas or summer* [laughs], *and you came.*" I think the change in weather and the change of people worked. So there's a treatment called traveling or seeing different people. It helped me, staying in Turkey for a few weeks with my family. Then I came back to Rochester.

"Maybe I will have my own company one day"

Family or other important points in your life are a good trigger to help you react to a crisis. Is that right?

Yes. Also, maybe distractions. I used to play squash once a week with a friend from Singapore. I wasn't competitive, and, at the time, my husband was making fun of my squash playing because I wasn't playing very hard. But I played every Friday with the same person for six years. It was so relaxing and distracting. So when you're in a PhD, trying to solve very difficult problems, it's important to let the mind get empty or calm. Either by going out of town, seeing family, or by playing a sport, or maybe an instrument… distractions!

Gulcin, it seems that your life is very good and you are very lucky. What you wanted to achieve, you are achieving. Is there any dream that you haven't fulfilled yet?

[*laughs*] I mean, maybe I will have my own company one day, which I never thought about. My dad had his own company for thirty years in the construction business, and my mom was a lawyer for thirty years. She had her own business. They always encouraged me to be independent. We'll see. I still love teaching, so I would love to teach at any place.

Probably it will be in California?

Oh, yes.

Where is home?

Home is here now: California! [*laughs*]

"Home is here now: California!"

...achieve driving proficiency based on real traffic situations.

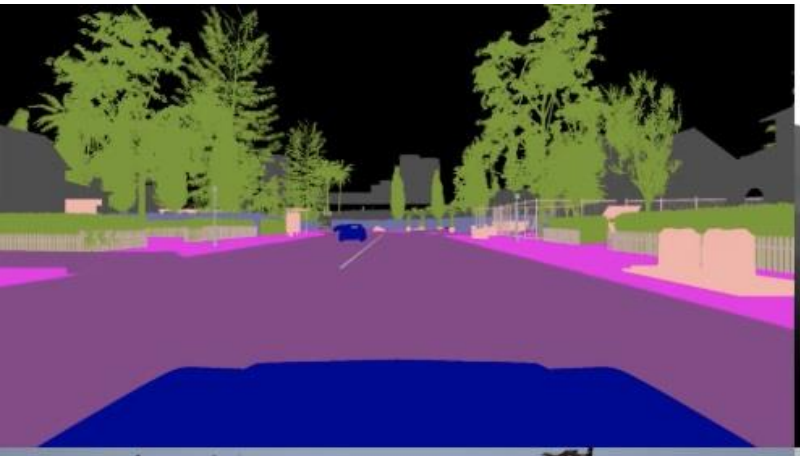
The CARLA Autonomous Driving Challenge aims to achieve driving proficiency based on real traffic situations. Each team needs to maneuver a self-driving car through the same set of predetermined routes. Altogether, the challenge consists of four parallel tracks, which differ in the type of input data provided to agents.

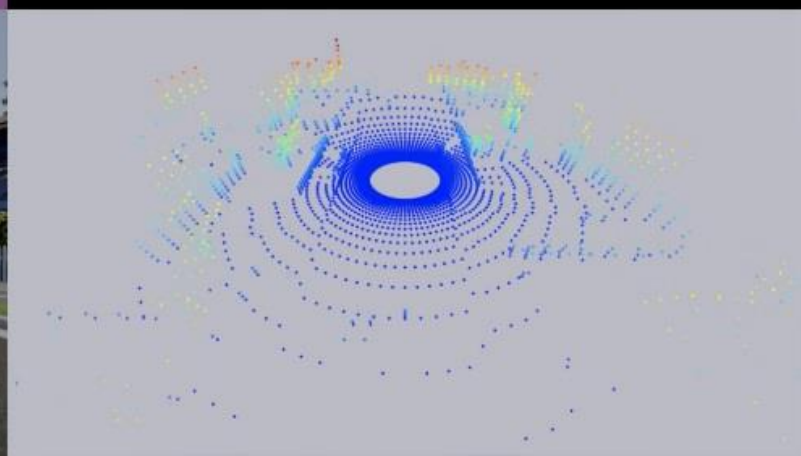
For each track, teams will need to follow specific instructions to reach destination points such as "GO-STRAIGHT, TURN-LEFT, or END" while navigating through intersections and roundabouts. Car, or "agents", will need to contend with real-life driving scenarios such as:

- Merging into a lane
- Changing lanes
- Navigating intersections
- Navigating roundabouts
- Following traffic lights and signs
- Yielding to pedestrians

The testing stage begins on May 20 and the challenge officially ends on June 1. Then CARLA will announce the winning team at the **CARLA Autonomous Driving Challenge Workshop**, to be held at **CVPR 2019** in Long Beach, CA.

Antonio Lopez, Associate Professor at the Computer Science Department of the Universitat Autònoma de Barcelona (UAB), works as one of the organizers of the CARLA Autonomous Driving Challenge. He shared with us captivating insight on the latest developments of the challenge.




Vladlen Koltun


Antonio M. Lopez


German Ros


Felipe Codevilla

On top: the co-organizers
Find out more at: carla.org
carlachallenge.org

"Everything is open and free!"

The project welcomes anyone to join. "Everything is open and free" explains Antonio. They expect a large number of participants and around 200 participants have already registered.

What Makes CARLA Significant?

CARLA utilizes sensors inside a simulator for conducting research on autonomous driving, including perception, planning, control, forecasting, etc. While many simulators exist, CARLA has some significant features. The project is open, not only in terms of code, but also regarding content such as the 3D models, desk tools, and materials. They aren't any restrictions on the use of CARLA. Without that, researchers in academia wouldn't have access to such a quality simulator. CARLA already has a community of users and support of start-ups offering services such as maps and models to support autonomous driving simulation.

Behind the CARLA Challenge

Vladlen Koltun's lab at Intel started CARLA at the end of 2016. At that time, the project required a mix of backgrounds: researchers, engineers, and artists.

Antonio Lopez recalls the early stages of the project which involved a collaboration between the Computer Vision Center (which he co-founded) in Barcelona, Vladlen Koltun's lab, and **German Ros**, the current Project Manager of CARLA. They took everyone's viewpoints to create the first version of the simulator.

This stage, besides including the software and the framework itself, also coincided with a research paper based on the simulator, illustrating how to train different models of AI for driving.

After more people joined the team, CARLA received additional sponsorship including from Intel and the Toyota Research Institute as well as from

General Motors Research & Development. They also gained support from KPIT. The Computer Vision Center (CVC) in Barcelona and the Open Source Vision Foundation also host CARLA.

Since CARLA can be run in the cloud, the challenge now has support from Amazon Web Services, in addition to Waymo, Uber, Audi Electronics Venture, Alpha Drive and EvalAI, who are also interested in the results of the challenge.

All members of the consortium have an interest in autonomous driving. CARLA has evolved to include new functionalities based on feedback from participants coming from academia as well as from industry.

Involving both sides of the community requires flexibility. They thought the best way of doing this was to propose a challenge. As people present different ideas, they can decide which ideas look more promising. Later, they can look at the results and see how the community responds.

Behind the CARLA Challenge

"The state-of-the-art of computer vision has been traditionally driven by challenges such as the PASCAL Visual Object Classes challenge, and the ImageNet Large Scale Visual Recognition Challenge. We work to make CARLA to play the same role for autonomous driving, at least when it comes to AI" explains Antonio.

The autonomous driving community has several ways of approaching the problem of self-driving cars. Some take a more traditional approach based on pipelines of different modules, for example, doing perception based on computer vision. This involves object detection and semantic segmentation which rely on neural networks. The results are given to a control module to maneuver the car locally so that globally it can move from a starting point to a destination.

However, this method requires annotating a lot of data to train the

neural networks. Antonio shares: "*There are other approaches called end-to-end driving. You drive your car. You collect perception data, like images, for instance. Together, with these images, you link some information from the car: the speed, the acceleration, the breaking. All of these measurements are acting as the ground truth; the perception data and this ground truth are used to train the AI driver.*"

"*In this case, you need very few manual interventions to train the networks, at least for annotating the data. You don't need to explicitly ask the network to detect anything like pedestrians, cars, or whatever is around you. You don't have this explicitly in the system. The system learns what is important and what is not important for driving. These are two very different paradigms*" he adds.

For the challenge, some teams follow the first paradigm, while others follow the latter. The results will reveal how these paradigms compete and what makes one better or worse than the other.

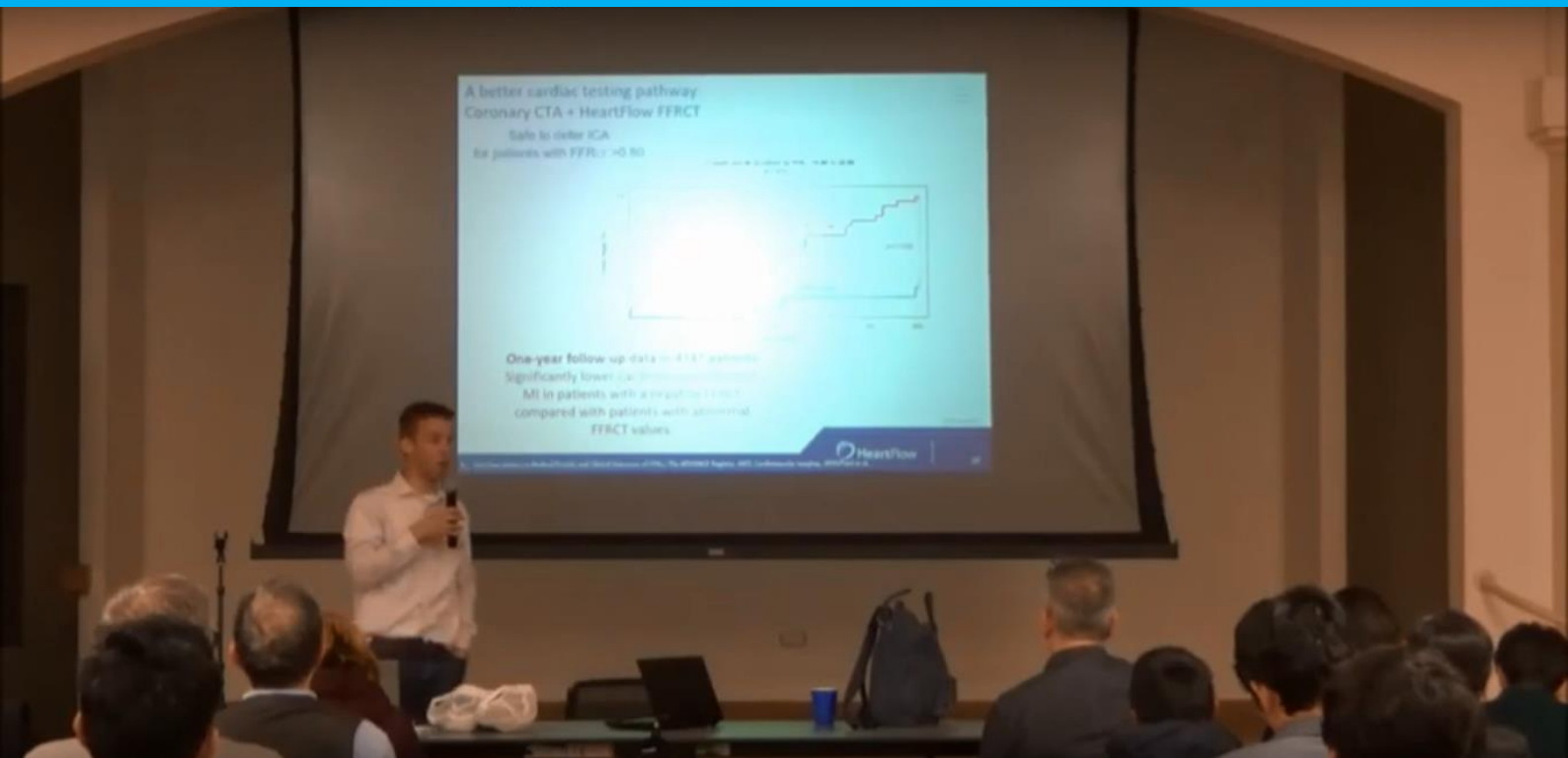
The challenge has four tracks, each with different difficulties. In one track, the team uses different sensors. In another, the team receives all the information so that people have a perfect perception, and they only need to focus on control.

Antonio elaborates: "*A novelty coming up now is more cities showing more complex scenarios like multiple lanes, with their respective individual traffic lights and roundabouts. Then cars have to negotiate intersections with other cars. They have to do maneuvers like overtaking another car. They have to take notice of pedestrians walking around. Also, from the very beginning of CARLA, we focused a lot on bringing variability. In CARLA, you have different light conditions and different weather conditions like rain… It's really challenging because there are a lot of variabilities that you can simulate.*"

Antonio has his own PhD students working on CARLA. So far, they made many unexpected discoveries, things they wouldn't have learned by processing a static dataset. He looks forward to seeing the end results:

"*I think the challenge is going to be great: we will see how CARLA can help the community to bring new knowledge about autonomous driving in different paradigms.*"

Among many other initiatives, RSIP Vision sponsors the Bay Vision Meetup. The April 4 meeting held in Cupertino focussed on the topic of AI in Medical Imaging with a great set of speakers: Aïcha BenTaieb (pictured above), Research Scientist at Roche, spoke about her work on "*Artificial Pathology - Computerized Deep Learning Models for Cancer Diagnosis*"; After Aïcha, Michiel Schaap, Director of Imaging Science at Heartflow, spoke about their analysis concerning "*Deep Learning in Clinical Practice*". If you missed the Meetup, both speeches are fully available in the video below!



Above, the third speaker: Mayank Kumar - Head of research at Gauss Surgical. Below is the truly yours, introducing the speakers. Why do I wear two hats? The answer is in the video of the Meetup, don't miss it! Join the Meetup

RSIP VISION

Global Leader in Computer Vision & Deep Learning

Women in Computer Vision

by Ralph Anzarouth

Women in Computer Vision (also called Women in Science) is a series of interviews conducted by Ralph Anzarouth. New interviews are regularly published on all RSIP Vision's publications: Computer Vision News and the Daily magazines (CVPR Daily, MICCAI Daily and many more).

Find now on the project page the direct links to almost 100 interviews... **at the click of a button**

Leadership

Mentoring

Competence

Confidence

Community

"The only way to succeed is to really start believing in yourself!"
Michela Paganini

"Most of all, you have to believe that you can do it!"
Laura Leal-Taixé

"It may look like a long list of names, but behind each name there is a fascinating world in which we were let in."
Ralph Anzarouth

Did you miss an interview? No worries, you can find them all in the **Women Scientist section of RSIP Vision's website**

FREE SUBSCRIPTION

Dear reader,

Do you enjoy reading Computer Vision News? Would you like to receive it **for free in your mailbox**?

Subscription Form
(click here, it's free)

You will fill the Subscription Form in **less than 1 minute**. Join thousands of AI professionals and receive all issues of Computer Vision News as soon as we publish them. You can also read Computer Vision News in PDF format (though this online view is way better) and visit our archive to find new and old issues as well.

We hate SPAM and promise to keep your email address safe, always.

Automatic Face and Gesture Recognition
Lille, France May 14-18 Website and Registration

Bay Vision Meetup: Artificial Intelligence in Medical Imaging
Cupertino, CA May 16 Website and Registration

ICRA - International Conference on Robotics and Automation
Montreal, Canada May 20-24 Website and Registration

RE•WORK - Deep Learning Summit (+ Healthcare)
Boston, MA May 23-24 Website and Registration

Digital Pathology & AI Congress: USA - 2019 meeting
New York City, NY Jun 13-14 Website and Registration

CVPR - Computer Vision and Patter Recognition
Long Beach, CA Jun 15-21 Website and Registration

CARS - Computer Assisted Radiology and Surgery
Rennes, France Jun 18-21 Website and Registration

AI & Big Data Expo Europe
Amsterdam, Netherlands Jun 19-20 Website and Registration

MIDL - Int. Conf. on Medical Imaging with Deep Learning
London, UK Jul 8-10 Website and Registration

IEEE ICCI*CC Int. Conf. on Cognitive Informatics & Computing
Milano, Italy Jul 23-25 Website and Registration

MIUA - Medical Image Understanding and Analysis
Liverpool, UK Jul 24-26 Website and Registration

Did we forget an important event?
Tell us: editor@ComputerVision.News

Did you read the Feedback of the Month?

It's on page 35!

FEEDBACK

Dear reader,

If you like Computer Vision News (and also if you don't like it) we would love to hear from you:

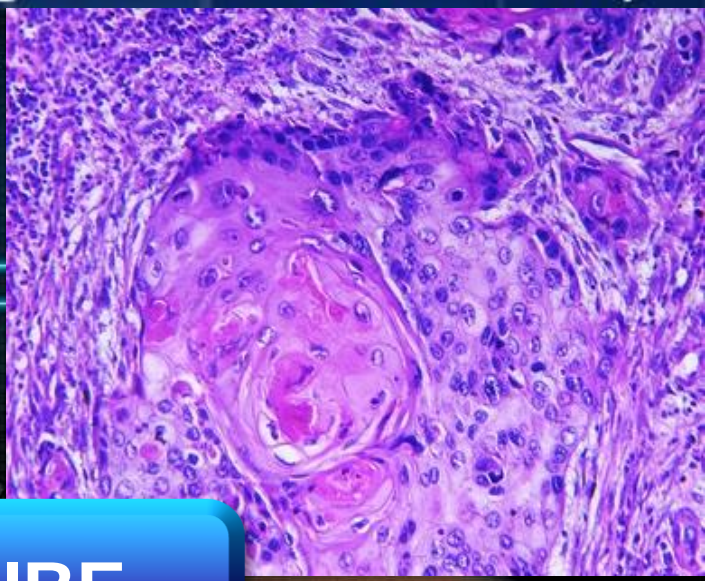
Give us feedback, please (click here)

It will take you only 2 minutes. Please tell us and we will do our best to improve. Thank you!

IMPROVE YOUR VISION WITH

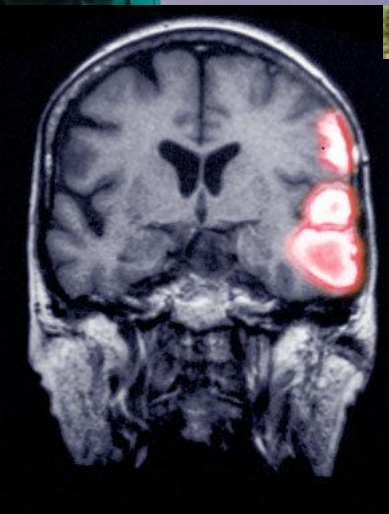
Computer Vision News

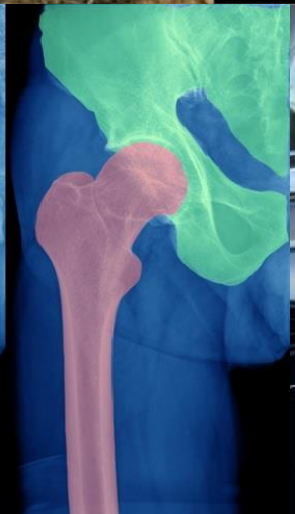
The Magazine Of The Algorithm Community



SUBSCRIBE

CLICK HERE, IT'S FREE





A PUBLICATION BY

RSIP VISION

Global Leader in Computer Vision and Deep Learning