Federal State Budgetary Educational Institution of Higher Education «Kazan National Research Technical University named after A.N. Tupolev–KAI»

### German-Russian Institute of Advanced Technologies

### Research in Computer and Systems Engineering

INSTRUCTIONS

to the practical work No. 3

by discipline "Basics of Neural Networks"

"Modern Convolutional Neural Networks. Computer Vision"

### Kazan - 2020

**PRACTICAL WORK No. 3**

**Name of practical work**

Modern Convolutional Neural Networks. Computer Vision.

**The goal of practical work**

To get understanding of convolutional neural network’s architecture and practical skills of image processing.

**TASK FOR PRACTICAL WORK**

1. Run a jupyter notebook file. *If you don’t remember how to do it, refer to Lab 1 document,* *paragraph 4.*
2. Start with file.

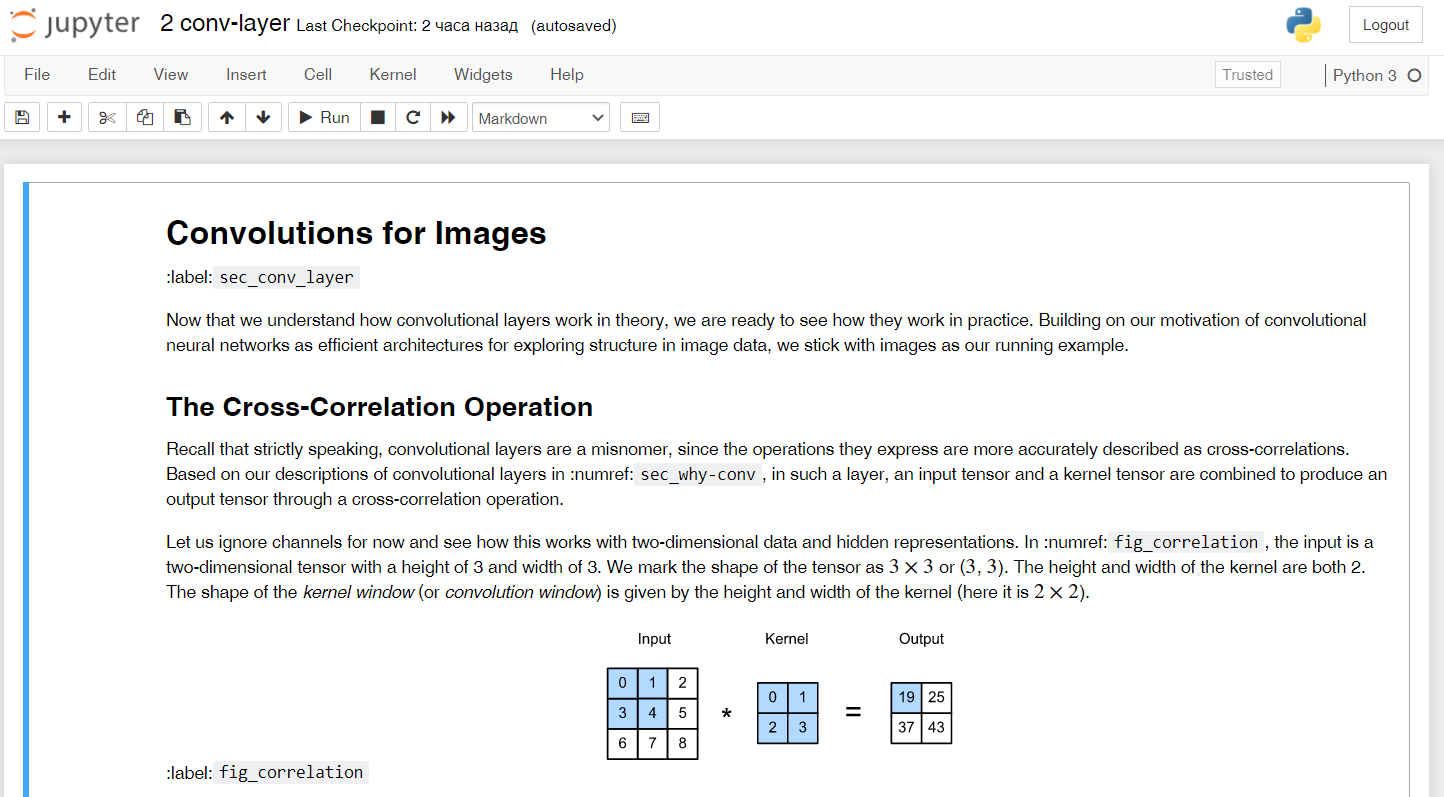


Fig. 1. Jupyter notebook with opened *2 conv-layer.ipynb*

Jupyter notebook consists of a sequence of cells. A cell is a multi-line text entry field, and you can execute its contents by clicking the Run button ( ) on the toolbar. Code cells can be run in any order you want. If the code takes too long to execute and you want to interrupt it, you can click the Stop button.

Further, to work with the code in this work, it is necessary to sequentially launch *every* block (even with text data only) using Run.

2. Sequentially Run the files in *convolutional-neural-networks* folder and after that in *convolutional-modern* folder.

**PRACTICE REPORT**

**Name of practical work**

Modern Convolutional Neural Networks. Computer Vision.

**The goal of practical work**

To get understanding of convolutional neural network’s architecture and practical skills of image processing.

**Done by**

Group № \_\_\_\_\_\_

Student’s Name, Last name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**The report is received in the form of an executable file for Jupyter Notebook or link to the Google Colab**

**Task:**

1) Setup the environment and run all the notebooks

2) Take dataset with images of the objects (on [Kaggle](https://www.kaggle.com/datasets), [COCO dataset](https://cocodataset.org/#home), [ImageNet](http://www.image-net.org/)). Train modern models for one object classification on the image (for example, dog). Compare the result on 5 different pictures on at least 3 different CNN architectures and fill the table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CNN/Recognition result (%) | Picture 1 | Picture 2 | Picture 3 | Picture 4 | Picture 5 |
| AlexNet |  |  |  |  |  |
| VGG |  |  |  |  |  |
| ResNet |  |  |  |  |  |
| …. |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CNN/Recognition time | Picture 1 | Picture 2 | | Picture 3 | Picture 4 | Picture 5 |
| AlexNet |  |  | |  |  |  |
| VGG |  |  | |  |  |  |
| ResNet |  |  | |  |  |  |
| … |  |  | |  |  |  |
| Picture 1 | | | Picture 2 | | | |
|  | | |  | | | |
| Picture 3 | | | Picture 4 | | | |
|  | | |  | | | |
| Picture 5 | | |  | | | |
|  | | |  | | | |

