* Видеокурс от Stanford University: <https://www.youtube.com/watch?v=vT1JzLTH4G4&list=PLC1qU-LWwrF64f4QKQT-Vg5Wr4qEE1Zxk>
* Курс от Физтех-школы Прикладной Математики и Информатики МФТИ: <https://vk.com/dlschool_mipt>
* Курс Нейронные сети и компьютерное зрение от Sumsung: <https://stepik.org/course/50352/promo#toc>

[Blur dataset | Kaggle](https://www.kaggle.com/datasets/kwentar/blur-dataset)

<https://colab.research.google.com/drive/1NwMieyhZh9td4KNtUMPKFr0yJ_ROYxCV?usp=sharing>

1. Collect and organize your dataset: Gather images that represent each of the categories you want to classify. It is important to have a sufficient number of images for each class. You can organize your dataset into separate folders for each category.
2. Split your dataset into training, validation, and testing sets: Divide your dataset into three sets: training, validation, and testing. The training set is used to train the model, the validation set is used to tune the hyperparameters and prevent overfitting, and the testing set is used to evaluate the performance of the model.
3. Preprocess your images: Preprocessing your images can help improve the performance of your model. You can resize your images to a consistent size, normalize the pixel values, and apply data augmentation techniques such as random rotation, flipping, and cropping.
4. Encode your labels: For multi-output classification, you will need to encode your labels for **each output separately**. For example, you can use one-hot encoding for the quality output and label encoding for the rotation and RGB-BGR outputs.
5. Load your dataset: Use a framework like TensorFlow or PyTorch to load your dataset into memory. You can use data loaders to efficiently load and preprocess your data in batches.
6. Define your model: Choose an appropriate architecture for your model, such as a CNN, and define the layers and parameters. For multi-output classification, you will need to define multiple output layers with appropriate activation functions.
7. Train and evaluate your model: Train your model using the training set and validate it using the validation set. Monitor the loss and accuracy during training and adjust the hyperparameters as needed. Evaluate the performance of your model on the testing set using appropriate metrics for multi-output classification.
8. Predict with new data: Once your model is trained, you can use it to predict the categories of new images. Preprocess the new images in the same way as your training data and use the model's predict function to generate predictions for each output.

**\*\*\*\* Toxic Comments Classification (Text Classification)\*\*\*\***

My client is an e-commerce retailer. They receive leads from different marketing channels such as email, telephone, website, messengers, shopping cart. Also, they collect leads as potential customers reviewing the comment they leave on the websites and e-commerce marketplaces.

My client asked me to help him to sort the incoming lead generation flow and classify the incoming leads to different groups in order to offer the groups different business proposals for the next step. The main task is to classify the highest potential leads and take them in the sales loop.

I have made some research, and it seems that the task looks like an natural language processing task, right? Could you please advise how it is better to execute this project, which machine learning algorithms to use and what the output can look like?