Catalyst workshop

DataFest 2020

Tl;dr - track organizer

Scitator

scitator.com

- Sergey Kolesnikov
- MSc in Math and Computer Science, 2018
- Interested in sequential decision making and lifelong learning
- RnD Lead @Tinkoff
- Researcher @MIPT
- Lead @Catalyst-Team
  - Alchemy - experiments logging & visualization
  - Catalyst - accelerated deep learning R&D
  - Reaction - convenient deep learning models serving

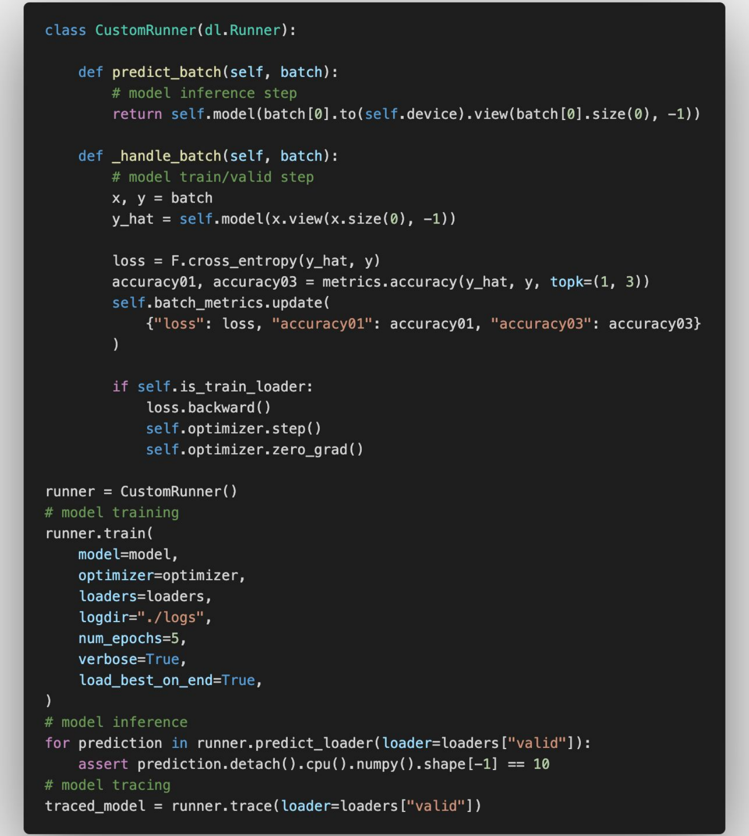
Catalyst

catalyst-team/catalyst

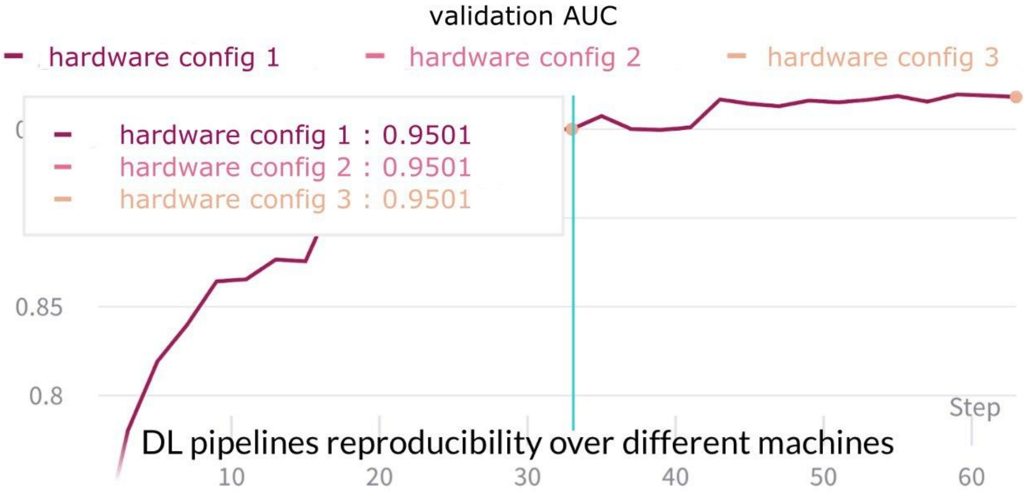
```
pip install catalyst
```

- High-level framework for deep learning R&D
- PyTorch Ecosystem
- Extra subpackages for CV and NLP
- catalyst-team.com

Catalyst

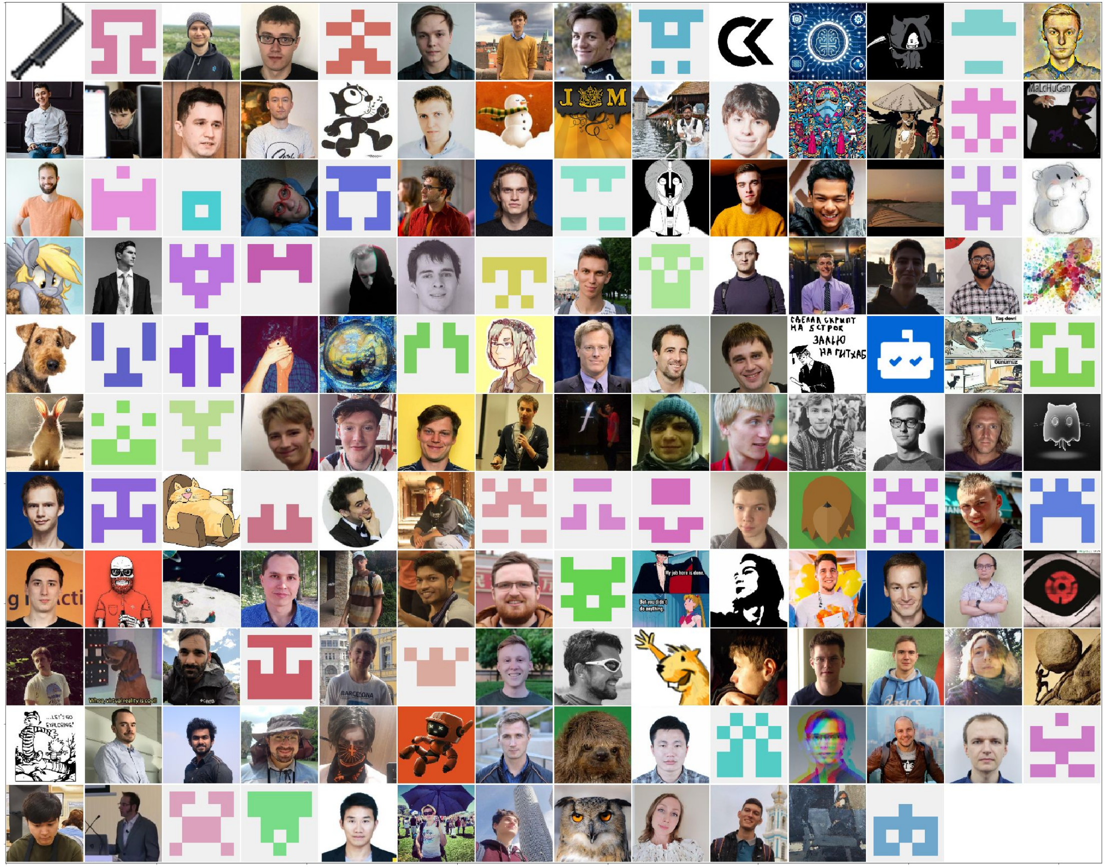


```
class CustomRunner(dl.Runner):

    def predict_batch(self, batch):
        # model inference step
        return self.model(batch[0].to(self.device).view(batch[0].size(0), -1))

    def _handle_batch(self, batch):
        # model train/valid step
        x, y = batch
        y_hat = self.model(x.view(x.size(0), -1))

        loss = F.cross_entropy(y_hat, y)
        accuracy01, accuracy03 = metrics.accuracy(y_hat, y, topk=(1, 3))
        self.batch_metrics.update(
            {"loss": loss, "accuracy01": accuracy01, "accuracy03": accuracy03}
        )

        if self.is_train_loader:
            loss.backward()
            self.optimizer.step()
            self.optimizer.zero_grad()

runner = CustomRunner()
# model training
runner.train(
    model=model,
    optimizer=optimizer,
    loaders=loaders,
    logdir="./logs",
    num_epochs=5,
    verbose=True,
    load_best_on_end=True,
)
# model inference
for prediction in runner.predict_loader(loader=loaders["valid"]):
    assert prediction.detach().cpu().numpy().shape[-1] == 10
# model tracing
traced_model = runner.trace(loader=loaders["valid"])
```

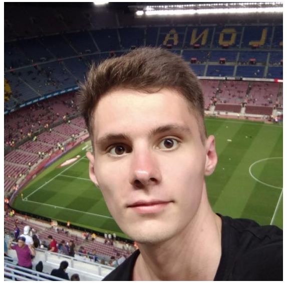


DL pipelines reproducibility over different machines

Team


MaLoHuGan
СДЕЛАЛ СКРИПТ НА 5 СТРОК ЗАЛИЛ НА ГИТХАБ
My job here is done.
But you didn't do anything
Whoa, virtual reality is cool!
...LET'S GO EXPLORING!

How Catalyst-powered projects advanced my academic career

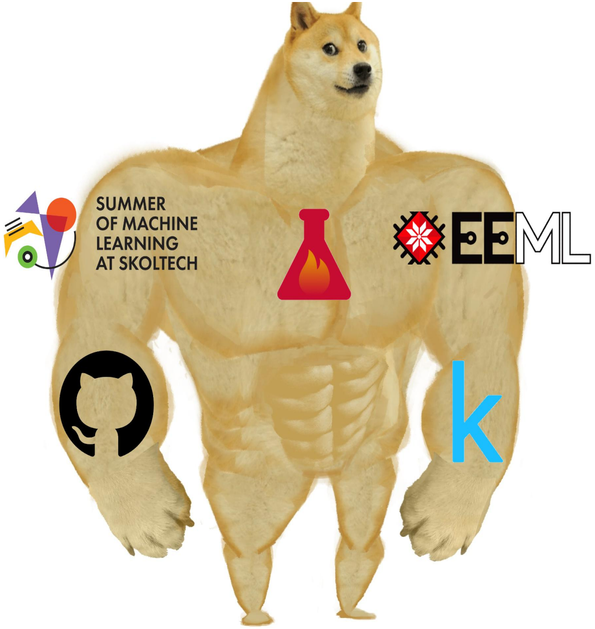


You probably have seen a lot of beginners in the field of deep learning struggle to write reusable well-structured code in their projects. Around a year ago I was at that point as well. Luckily I bumped on a Kaggle kernel which was using a Catalyst framework and it really encouraged me to dive deeper in Catalyst and apply its best practices in my own projects.

Very soon I earned a bronze medal in semantic segmentation competition. After the competition has ended I decided to write a small research paper describing my approach. This allowed to participate in two machine learning summer schools this year. At the same time I was getting better in Catalyst and DL and won a silver medal in another computer vision task. After the finish of the second competition I was able to contribute to Catalyst library by implementing a callback which applies a Cutmix data augmentation.

As a result, not only I learned how to write deep learning pipelines, but also built projects which allowed me to participate in different academic events and get noticeable lines in my CV.

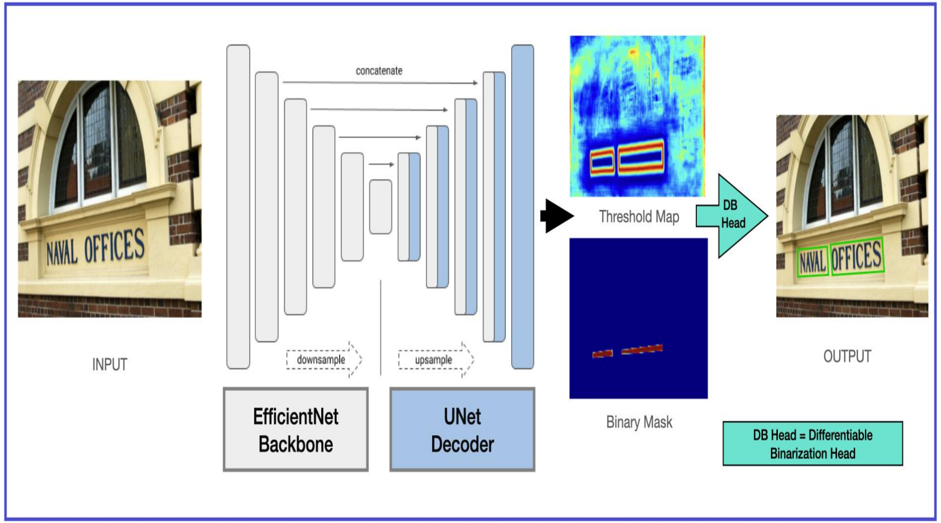
Kirill Vishnyakov (Cyr1ll)



Multilingual Text detection in Video/Images

Among the pattern recognition fields, automatic text recognition, known as OCR, has been widely studied for its prominent position in our everyday life. OCR has a long history of research that started from isolated character recognition and evolved to printed/ handwriting document recognition.

Lokesh Nandanwar (lokeshkvn)



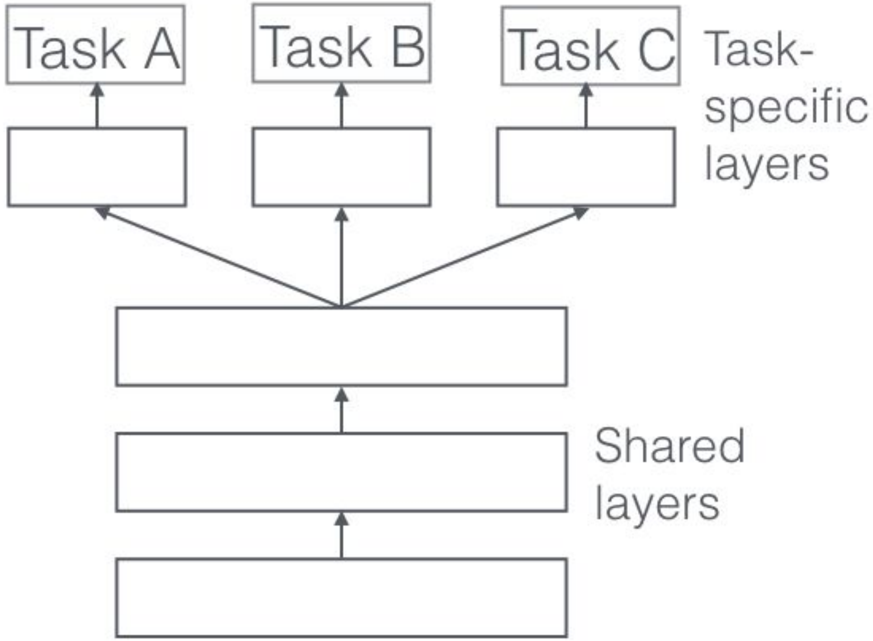
Over the last decade, interest in this area of research has led to a plethora of text detection and recognition methods. So far, these methods have focused only on some languages such as English, Latin and Chinese. For a language like Hindi, Russian or Arabic which is also used by more than one billion people around the world, the literature is limited to very few studies. This presentation aims to tackle the challenges in Multilingual Text detection in Video/Images with the help of Catalyst framework.

Multi-task learning in the real world with Catalyst

Hristo Vrigazov (dnn.cool)

Many real-world problems can be solved very computationally cheaply using multi-task learning - where we share weights between tasks and have task-specific layers only in the final few layers of the neural network. But the complexity of managing a multi-task learning project grows very quickly.

Using Catalyst makes such a project much easier in the real world, and we will have a look at three different use-cases and how Catalyst was used to greatly ease the training, evaluation, interpretation and tuning of the results in a multi-task setting.



Firing a cannon at sparrows: BERT vs. logreg

Yury Kashnitsky (yorko)

There is a Golden Rule in NLP, at least when it comes to classification tasks: “Always start with a tfidf-logreg baseline”. Elaborating a bit, that’s building a logistic regression model on top of tf-idf (term frequency - inverse document frequency) text representation.

This typically works fairly well, is simple to deploy as opposed to neural net and that’s what already deployed and working day and night while you are struggling with fancy transformers. In this presentation, we will go through a couple of real-world text classification problems and speculate on the reasons to resort to BERT as opposed to good old tf-idf & logreg.

Meanwhile, we will discuss a Catalyst text classification pipeline with HuggingFace.

Pre-training Transformers with Catalyst

Anatoly Potapov (anatolii)

Since the NLP's Imagenet moment (emergence of ELMO, BERT), language models are widely used as a backbone for a variety of supervised tasks (intent classification, named entity recognition, question answering), etc

At Tinkoff, we have tens of millions of unlabelled customer conversation samples. In such a scenario, it is highly beneficial to pre-train on in-domain data and having a custom vocabulary. We transferred our pre-training pipeline to the Catalyst framework and reduced our codebase while keeping features like distributed training and fp16 training.

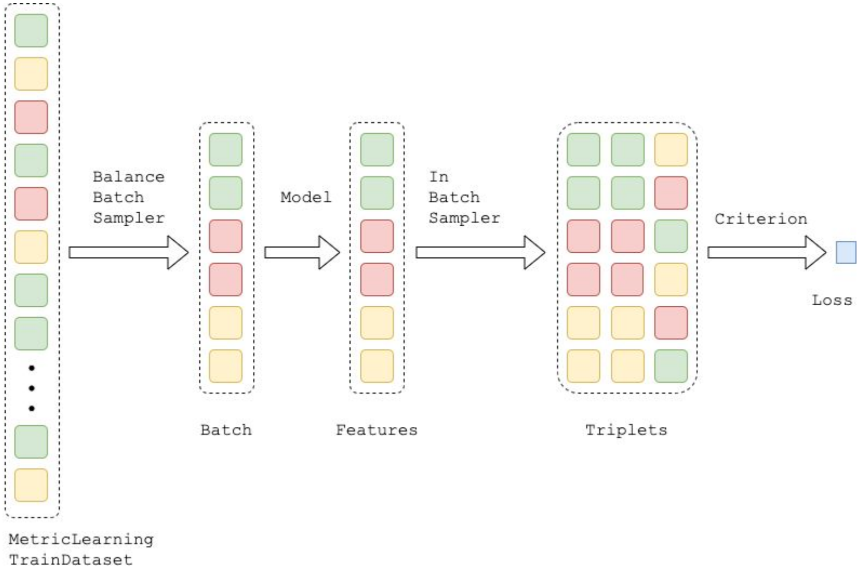
In my presentation, I will tell you how to pre-train transformers at scale with the Catalyst framework without writing lots of "infrastructure" code.

Metric learning with Catalyst

Aleksei Shabanov (Bender)

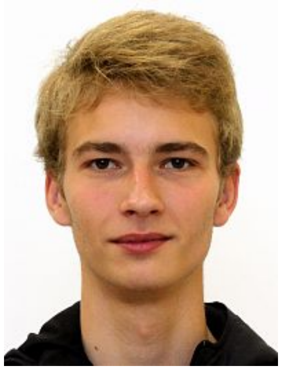
Let's imagine that you need to map your data points (images, texts, sounds and so on) to some vectors. For instance, in the case of working with images, the simplest solution would be to take the outputs of the penultimate layer of classifier. But what if you want to do something smarter?

To solve such a problem, you can use set of tools from the field of Metric learning: sample triplets of data points using various strategies, use triplet loss, and calculate retrieval metrics. We have implemented all these tools for you in one place and provided a user-friendly interface so that you can easily adapt it to your task. In our video section, we will show with specific examples how the work with Metrics learning pipeline is arranged in Catalyst.



Pruning and Lottery Ticket Hypothesis with Catalyst

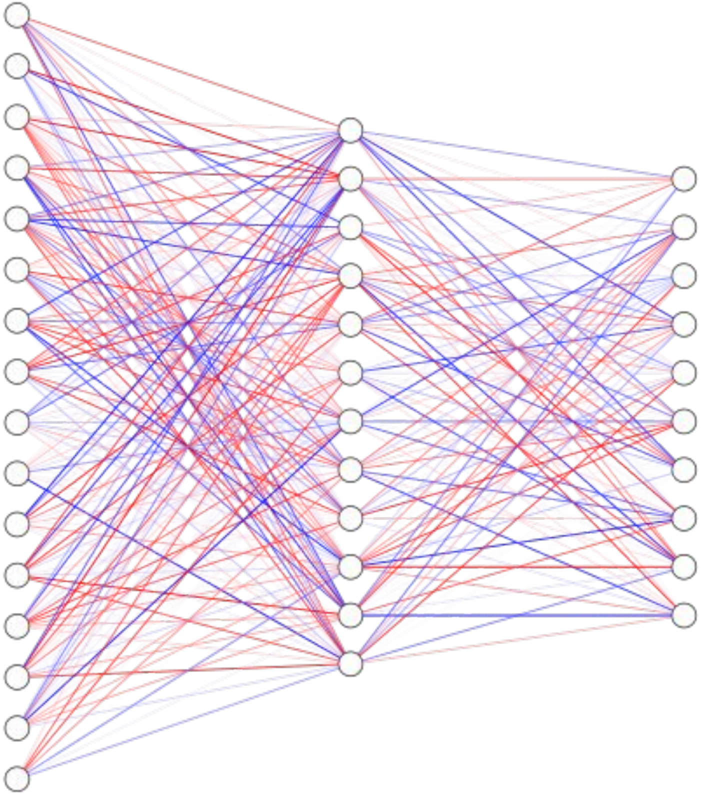
Nikita Balagansky



In the past few years, state-of-the-art architectures become more and more complex the number of parameters grows exponentially. But what if all networks are over-parametrized and more than a half parameters don't influence the result. The process of removing connections between neurons called pruning.

Moreover, some examples came from the biology field. For instance, the human brain is also over-parametrized in the first stages of growth, and we are learning through pruning unnecessary connections.

In my presentation, I will tell you how to perform pruning of your network with the Catalyst framework and will briefly demonstrate what is "Lottery Ticket Hypothesis" mean.

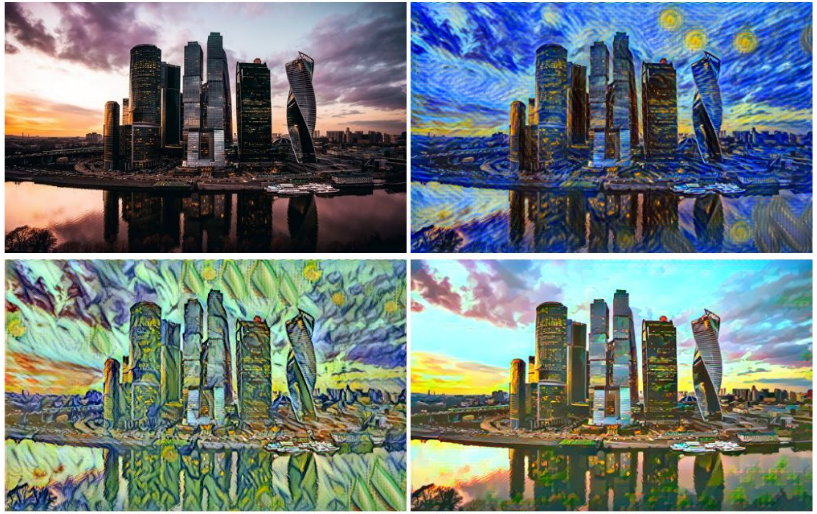


Style transfer with Catalyst

Ivanov Aleksandr(inkln)

There is no really good metric in the task of transferring style from one image to another in real time. The main popular metric is human opinion. To achieve the best result, you need to iterate over many hyperparameters, train models, and submit the results for evaluation to a person.

The ability to customize the experiment provided by the Catalyst allows you to train many models with small differences in hyperparameters at the same time, and then evaluate the results. The process of setting up the next experiment becomes instant, and high-quality logging allows you not to get confused in the trained models. You just have to choose the most pleasant one.



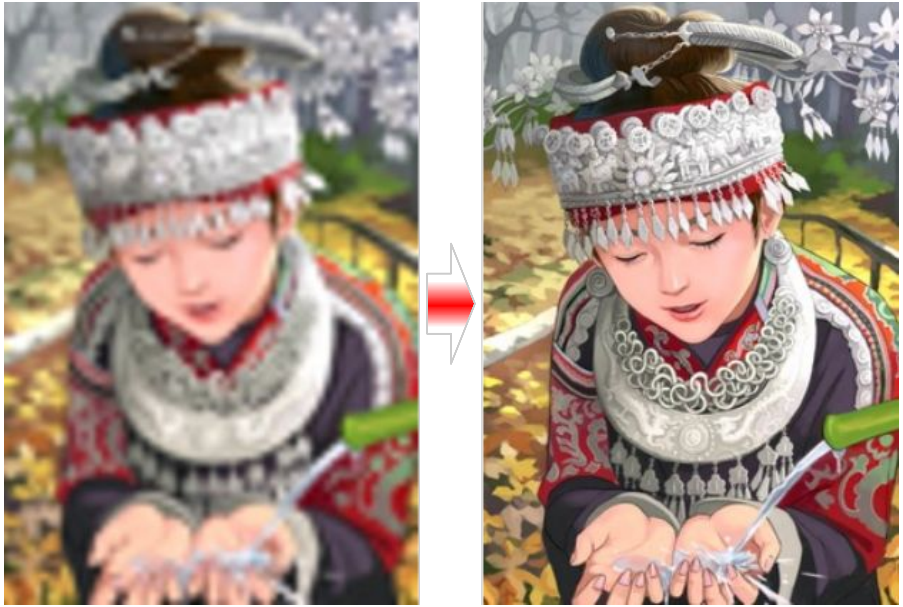
Super Resolution with Catalyst

Yauheni Kachan (bagxi)

You probably have seen a lot of CSI movies in which zooming and enhancing of images are using to identify a person or recognize number plate on the photo of low quality.

That's true that reality differs from cinematography, but it doesn't mean that only big brother is able to use such technology. Furthermore, with the help of Catalyst, it is possible to make your own model (ESRGAN to be more specific) to upscale images several times just in one bash command.

Sounds interesting? Join the Catalyst section!



Catalyst

catalyst-team/catalyst

pip install catalyst

#tool_catalyst

@catalyst-slack

@catalyst_team

@catalyst_core