Note on train and inference

Unfortunately, I could not neither fine-tune nor train models all by myself. I used already pretrained models, because research in that direction takes more hours than I actually have available. Despite that I have tried to train from the scratch and run into some troubles building MagicPoint architecture and training script. The main reason for that is lack of time for troubleshooting.

Also my algorithm can extract points of interest from the satellite image, but inference for now can only be performed only on regular images (of size 480x480). Considering size and amount of key points my current work station will blow up, if I tried to match points on satellite imaginary. There is another problem arises – visualization.

Enhancements

Despite the fact that I did not train the algorithm I can forecast some changes to it, as during my time testing written code I noticed some flaws. Let's look at what would I have done:

1. **Perform fine-tuning**. Yep, this states the obvious, but still must be mentioned. It is the main improvement that can be done for now. SuperPoint must be fine-tuned because angle has changed, the size of the objects also has changed, and I guess there were not similar satellite imaginary data in the MS COCO 2014 dataset. I also think that maybe SuperGlue also needs to be fine-tuned, because the number is too high to easily overwhelm a neural network that did not encounter such an amount before.
2. **Additional random transforms.** Currently I am extracting images from the Google Earth Pro, which is a great resource for high-quality images of the Earth surface. From those images I generate a dataset by splitting 8160x8160 images into 480x480 images and for each of the child images I generate an augmented image and a mask. Unfortunately, for now augmentation is limited to random rotation transform, which is not great if we want this algorithm to be robust. To make the dataset more diverse and closer to a real-world dataset, I would have also added: *random noise*, *random scale*. I am currently thinking of random perspective transform, but I don't think it would be useful, because data from the satellite can only be of one point of view.

3. **Winter imaginary.** GEP is a great resource, but it does not contain any images with snow. This limits our dataset to summer, autumn and spring. To make the algorithm more robust we need to add pairs with non-winter/winter images.