**Disparity Map Elaboration Using**

**CNET Neural Network**

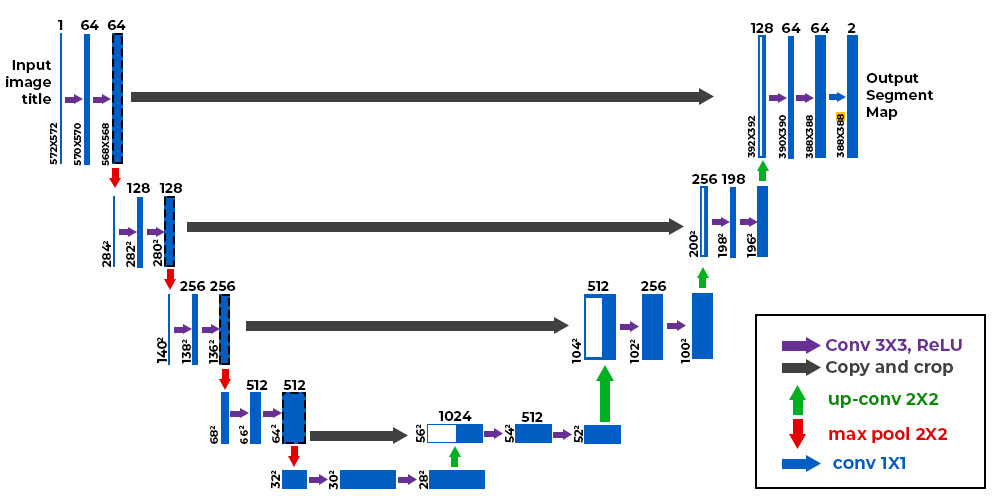
In the following paper is illustrated the attempt of producing a *Disparity Map,* utilizing a neural network, from a single image input.

Usually disparity maps are produced by finding the differences in stereo images taken by two nearby cameras; the result is an image that shows, using shades of gray, the disparities of the images and as such the depth of the various elements.

The necessity of using two cameras is an inconvenience because they need to be always aligned with each other and at the same distance, so the whole device sometimes is too unstable to be used in daily applications.

To overcome these issues it has been tested the possibility of using only one camera to acquire images and then pass them to a convolutional neural network to try to generate a disparity map.

In this approach it has been tried the *U-NET* model, usually used for image segmentation and as an underline of the *Stable Diffusion* model.



The model has been tested using rgb and grayscale images with grayscale ground truth as training input and generating a grayscale disparity map output.

For this test it has been used a dataset composed by around 700 indoor images of daily objects, splitted into train and test sets, and then resized and normalized to boost the performance of the network and then passed to the network in batches of 32.





Top image: Original (1080x860), Bottom image: Ground Truth (1080x860)

There hasn't been a relevant difference in the using of rgb images instead of grayscale images, so it has been decided to go with the second approach to reduce time to train and evaluate the model.

The results of the experiment were in the expectations, producing low accuray images in respect to the ground truth but still promising considering the low computational power and the relatively small dataset.

The model generates a recognizable pattern but not accurate enough to produce a reliable disparity map.

Another hyperparameter evaluated in the train of the network was the use of various loss functions to better converge to an accurate solution.





Top image: BCEWithLogitsLoss (256x256), Bottom image: MSELoss (256x256)

It has also run a long test tripling the training epochs and reducing the image size to 128x128; that produced better visual results with higher accuracy overall.



30 Epochs, BCEWithLogitsLoss, (128x128)

The experiment was also meant to set the ground for future iterations, with the perspective of trying to use this type of network to produce accurate results by utilizing directly stereo images or other types of images where depth can be extracted with traditional approaches.