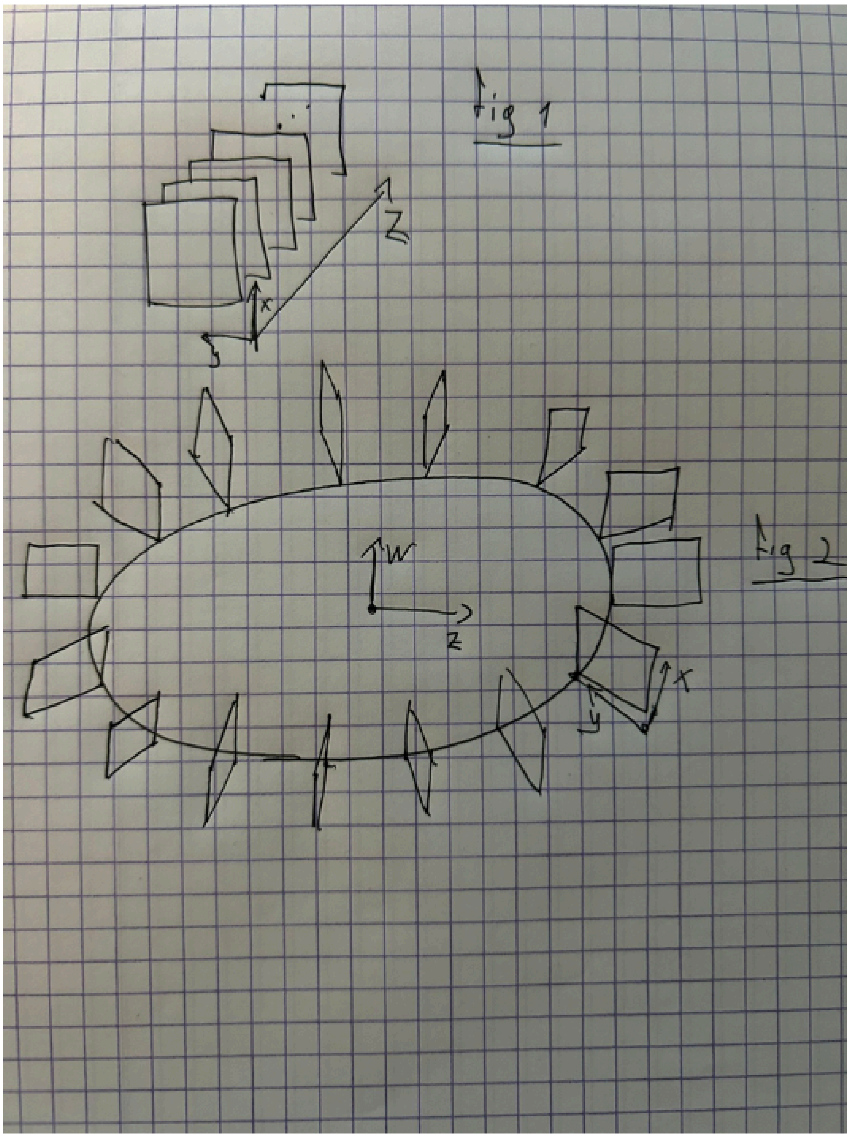
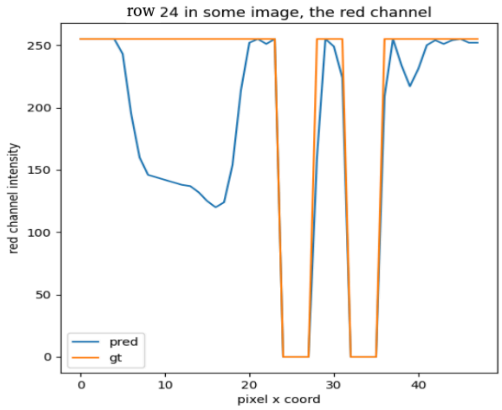
Image Embedding

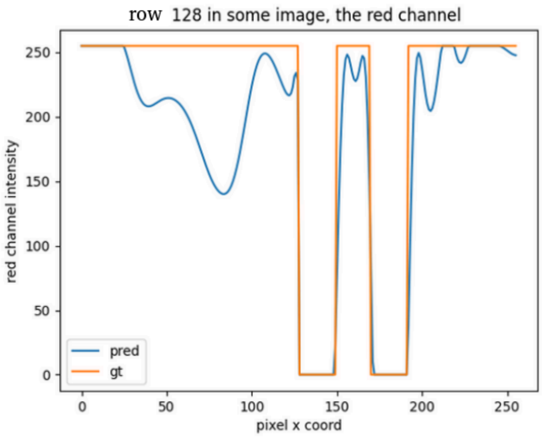
On how to choose the architecture -

In order to learn more images, I needed to add at least one more dimension to the data - the Z axis.Now, the actual range of our function is effectively 3 dimensional (as seen in fig 1.)
And yet, if we would like to interpolate between images, the interpolation path will traverse through other images from the set. This would never allow us to interpolate between images that are not consecutive in the set. meaning, not without going through all other images in between. For this reason I have chosen to move one dimension up and embed the images on a ring at the WZ plain(as seen in fig 2).



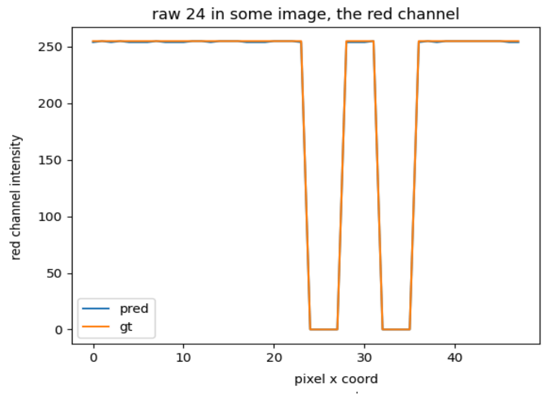
One more thing worth noting, is the way to qualitatively assess the nn ability to reconstruct the images. It is done by visualizing an image single channel's row (blue -Siren function, orange -GT)

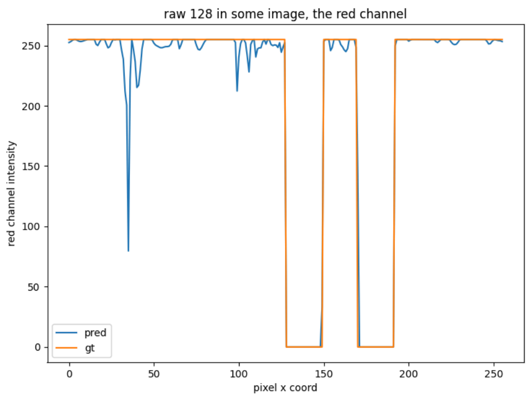




This is an example of a bad reconstruction(left), and its upsample to a resolution of 256 pixels.

In order to choose a proper architecture for the nn the question was- "Is dipper and wider, are necessarily better". My initial guess was *no*. As we wish to upsample the image, too dip could mean that, apart from training being expensive and long, that there would be a degeneracy in the solution. let's have a look at such a case -





here we can see that though the nn have a good fit w.r.t. the data still the Siren function in essence converges to a bad solution. In our context this is an over fit.
and this is how it looks as an image(left is the reconstruction, right is the GT) -
the added lines are the high frequencies artifacts.

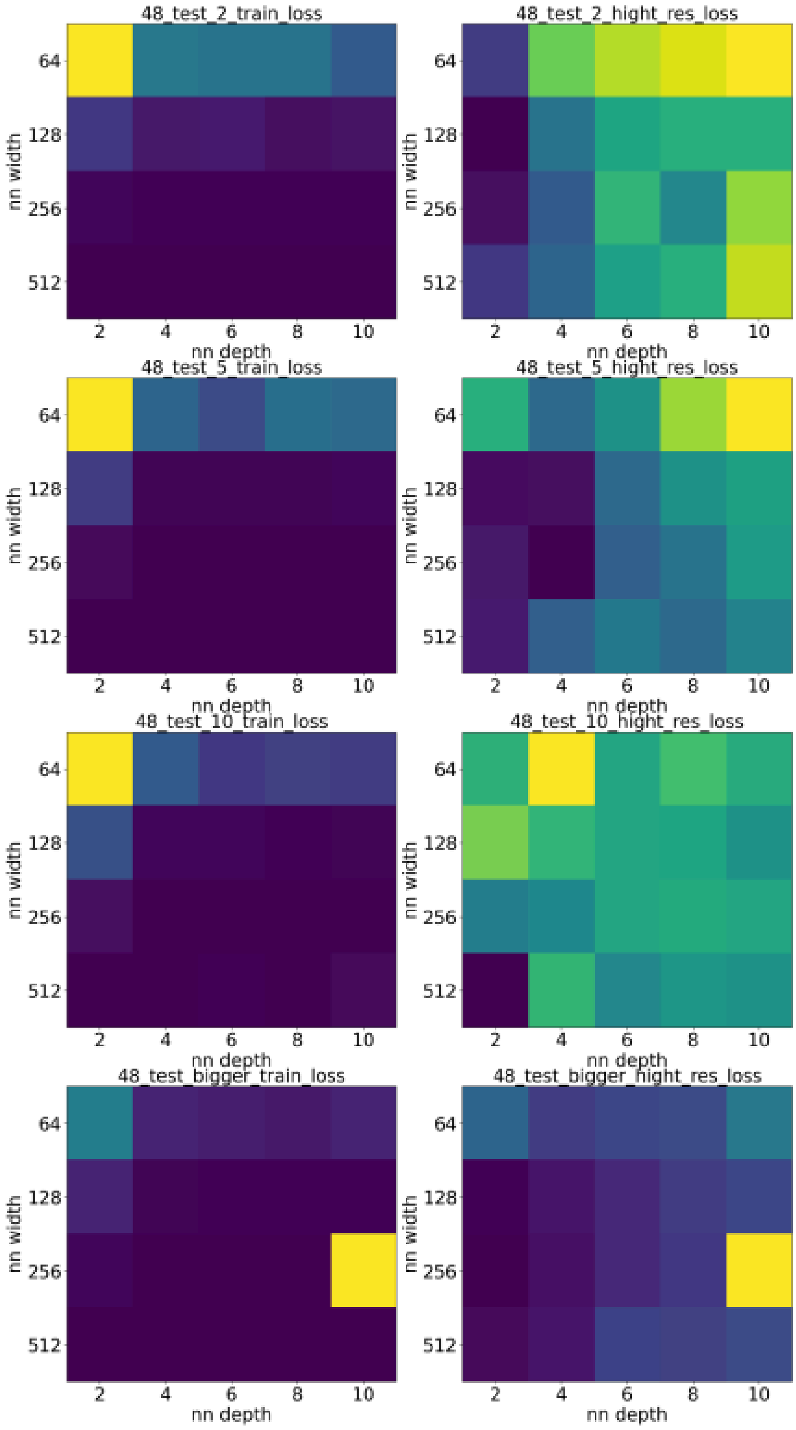
i) question - Define what generalization means in this specific task. Are there several kinds

For this task one form of generalization would be to be able to up sample.
and since the I am adding more coordinates in order to embed the images, generalization could mean that the space inbetween the images behaves well, but we will address this issue in 2b

ii) question - Track the network's generalization during training

Grid search -

in the following figure is a fit over several data sets of sizes [2, 5, 10, 15] and their train MSE vs their test MSE for 600 iterations -

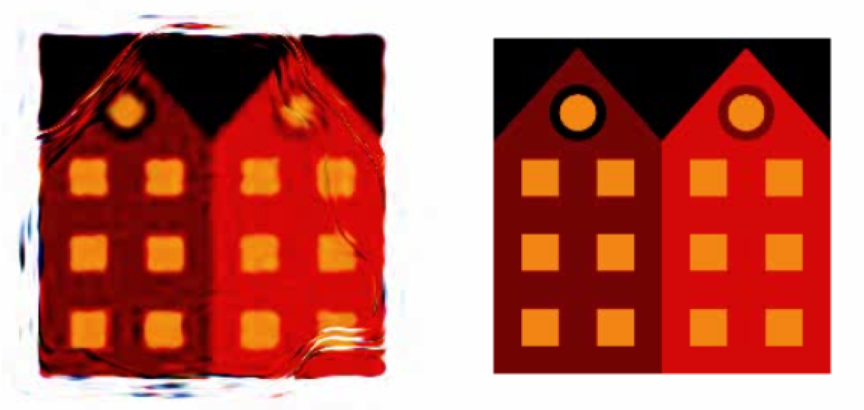


Let's try and get the feel of what is a good depth and width of the Siren function w.r.t some data size. where depth is the number of hidden layers and width the number of activations at each layer.
to do so we could just do a grid search over the full data set and find the best solution.
Yet determining the grid, by itself is a difficult task as any mistake could be very expensive, both in research time and resources. It is best to draw conclusions from a miniature example.
From this grid search It seems that dipper is not necessarily better, but, wider could be helpful.

Image interpolation - Generate new images.

a) First, demonstrate the networks ability to upsample the images.

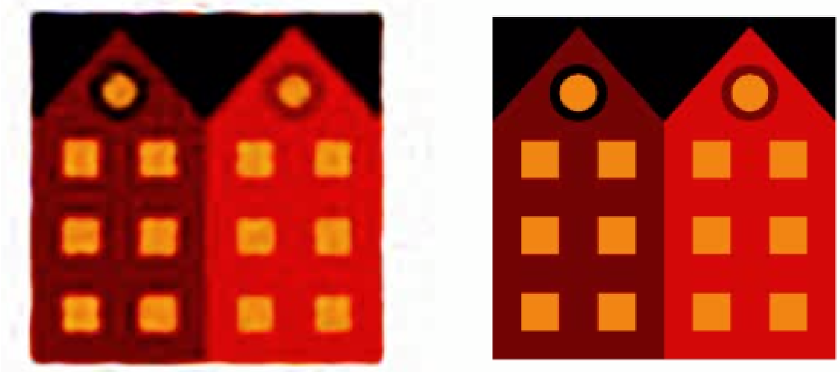
Running the entire data (before running the grid search) with a nn of depth 9 and width 512, I receiving the following image-



it seems that we are dealing with an over fit and it would be better to lower the nn depth.

i)Upsample the images to a resolution of 256x256x3.

Running with depth -4 and width 512, yielded better results -



b. Next, interpolate between selected pairs of images:

i) Consider known similarity measure between representations, and describe one that you think is helpful for the following

Earth movers distance (EMD) will measure the minimum amount of work required to transform one histogram into another I suggest the following normalized measure-

$$\frac{EMD(I_1, I(\alpha))}{EMD(I_1, I_2)}$$

- where I(alpha) = I1*alpha + I2*(1-alpha)

for alpha = 0 it should be 0 and for alpha=1 it should be 1.
and for a well behaved function that interpolates well this measure should move from 0 to 1 smoothly not exceeding 1 to much.

ii) Select (three) pairs of images which will be "best" for interpolating between the images (in each pair).

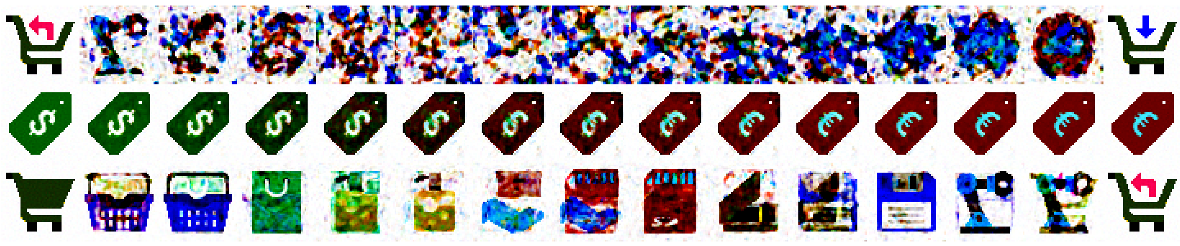
I have chosen the pairs -

- buy - return_purchase
- price_tag_euro - price_tag_usd
- return_purchase - shopping-cart

they appear at the edges of the interpolation scheme(the figure below).

The first pair are very far in terms of location in the data set, the second pair are adjacent in the data, and the third pair is 8 images away from one another.

iii) Demonstrate interpolation between the two images in each pair selected. Repeat this for the three pairs of images.

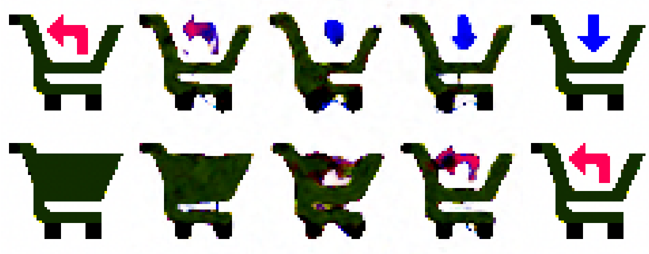


vi) Describe any shortcomings of your results, and hypothesize about why they occur

- first row - diverged completely,
- second row - good interpolation
- third row - goes through all images that are between source and target in the dataset,

The third row is a clue to what is happening in the space where there is no signal to regulate the Siren function.

looking at smaller data sets it seems that this method can in fact converge -



Hypothesis -
Our source data looks kind of like fig 1 (after projection from 4D space where it actually exists). The colored rectangles represent images in the set.
Fig 2 represents the Siren function; colored points on the circle represent reconstructed images. note that in this scheme every point is in the WZ plain and represents a whole image. The colored regions represent points that resemble one of the images from the set according to the color.
Also exists a region where the Siren function has diverged (the black circle). and is far away from resembling any image

fig 3

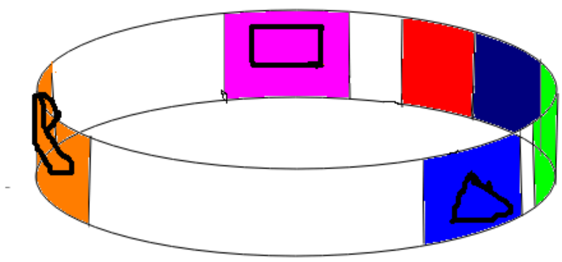
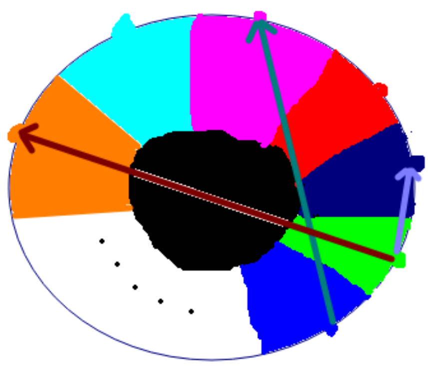


fig 2



Improved image interpolation

my solution is inspired by inhomogeneous Helmholtz equation

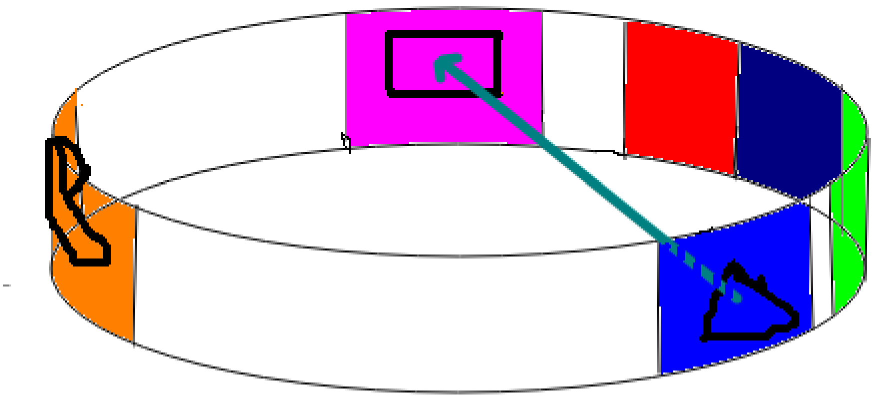
https://en.wikipedia.org/wiki/Helmholtz_equation#Inhomogeneous_Helmholtz_equation

$$\nabla^2 A(\mathbf{x}) + k^2 A(\mathbf{x}) = -f(\mathbf{x}) \ \text{ in } \mathbb{R}^n,$$

instead of using the exact formulation of the Helmholtz equation I'll use a concept simpler for implementation.

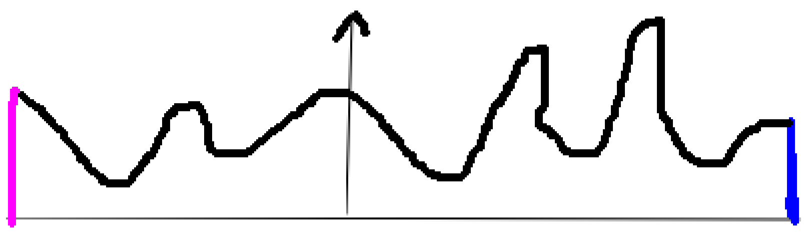
One of siren's most esteemed features is the ability to estimate the laplacian of the Siren function. In our current situation there is no regularization of the function inside the unit circle. This would mean that on the green line plotted in fig 3, there is no regularization.

fig 3

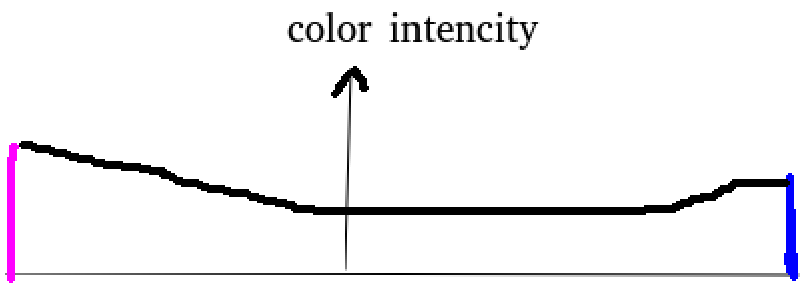


on that green line anything can happen except on the edges. thay are regulated by the image as shown below.

color intencity



we would like instead for the transition of color to move in a more “relaxed” manner -



do to so I propose a regularization of the laplacian.

$$\sum_{i=1}^{n} \nabla^2 \phi(x, y, z(\alpha_i), w(\alpha_i))$$

As the Siren function is smooth, adding this regularization across randomly sampled points within the unit circle would encourage the Siren function not to diverge and to move smoothly from one image to the other.