

## Vaja 2

Pri tej nalogi smo implementirali upscaling resolucije slike s pomočjo NM. Prikazani primeri so bili poginani na cpu napravi zaradi slabe podpore AMD grafičnih enot in ML.

### **Podatkovni set Set14:**

Average height: 445.571429

Average width: 491.500000

Height interval: [276, 656]

Width interval: [250, 768]

Number of images: 14

### **Podarkovni set Set5:**

Average height: 336.000000

Average width: 312.800000

Height interval: [256, 512]

Width interval: [228, 512]

Number of images: 5

### **Podatkovni set T91:**

Average height: 203.582418

Average width: 264.120879

Height interval: [78, 387]

Width interval: [78, 508]

Number of images: 91

=====

Layer (type:depth-idx)	Output Shape	Param #
------------------------	--------------	---------

=====

SuperRes	[1, 1, 33]	--
└─Conv2d: 1-1	[64, 1, 33]	5,248
└─ReLU: 1-2	[64, 1, 33]	--
└─Conv2d: 1-3	[32, 1, 33]	51,232
└─ReLU: 1-4	[32, 1, 33]	--
└─Conv2d: 1-5	[1, 1, 33]	801

=====

Total params: 57,281

Trainable params: 57,281

Non-trainable params: 0

Total mult-adds (Units.MEGABYTES): 65.21

=====

Input size (MB): 0.00

Forward/backward pass size (MB): 0.03

Params size (MB): 0.23

Estimated Total Size (MB): 0.26

=====

### Parametri:

```
model = SuperRes()
optimizer = torch.optim.Adam(model.parameters(), lr=0.0001)
loss_fn = nn.MSELoss()
epochs = 5000
num_images = 32
```

### Predobdelava:

```
def calculate_frame_offsets(dirpath):
    offests = np.empty((0,3), dtype=np.int32)
    dirlist = os.listdir(dirpath)
    for n in range(len(dirlist)):
        img = cv2.imread(dirpath+dirlist[n])
        for i in range(0,img.shape[0]-32,14):
            for j in range(0,img.shape[1]-32,14):
                offests = np.append(offests,[[n,i,j]],axis=0)
    return offests
```

### Training DataSet Generator s predobdelavo:

```

class ImageDataSet(Dataset):
    def __init__(self, dirpath):
        self.dirpath = dirpath
        self.offsets = calculate_frame_offsets(dirpath)

    def __len__(self):
        return len(self.offsets)

    def __getitem__(self, index):
        dirlist = os.listdir(self.dirpath)
        img_orig = cv2.imread(self.dirpath+dirlist[self.offsets[index][0]])
        scale = random.randint(2,4)
        img = cv2.resize(img_orig, (int(img_orig.shape[1]/scale), int(img_orig.shape[0]/scale)))
        img = cv2.resize(img, (int(img_orig.shape[1]), int(img_orig.shape[0])))
        img = cv2.cvtColor(img, cv2.COLOR_BGR2YCrCb)
        img_orig = cv2.cvtColor(img_orig, cv2.COLOR_BGR2YCrCb)
        img = img[self.offsets[index][1]:self.offsets[index][1]+33,self.offsets[index][2]:self.offsets[index][2]+33,:]
        img_orig = img_orig[self.offsets[index][1]:self.offsets[index][1]+33,self.offsets[index][2]:self.offsets[index][2]+33,:]
        img_orig = img_orig/255
        img = img/255
        return np.array(img).astype(np.float32),np.array(img_orig).astype(np.float32)

```

## Rezultati:

Set14:

PSNR LR: 31.851712

PSNR HR: 31.461059

SSIM LR: 0.728591

SSIM HR: 0.731238

Set5:

PSNR LR: 34.000212

PSNR HR: 33.892944

SSIM LR: 0.882290

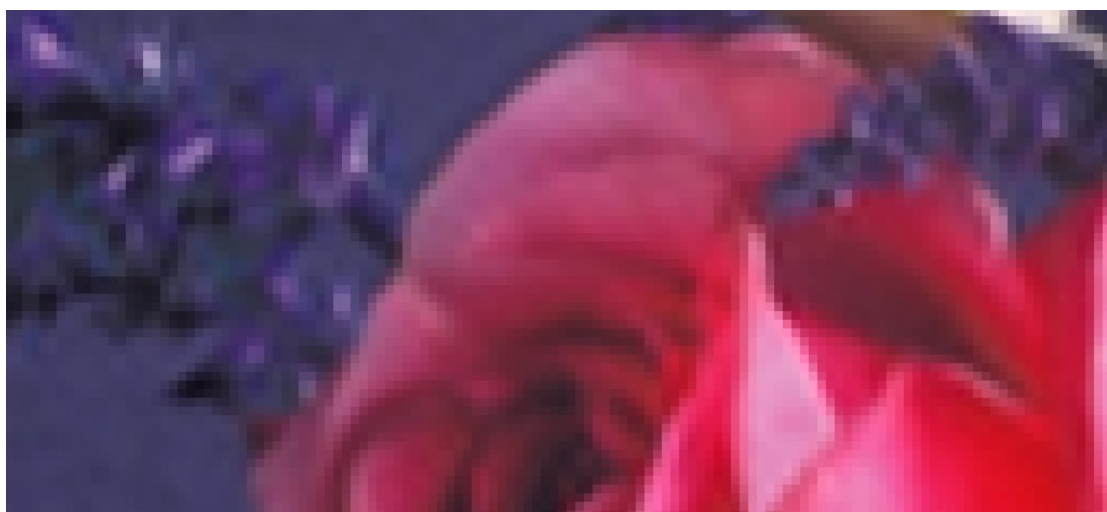
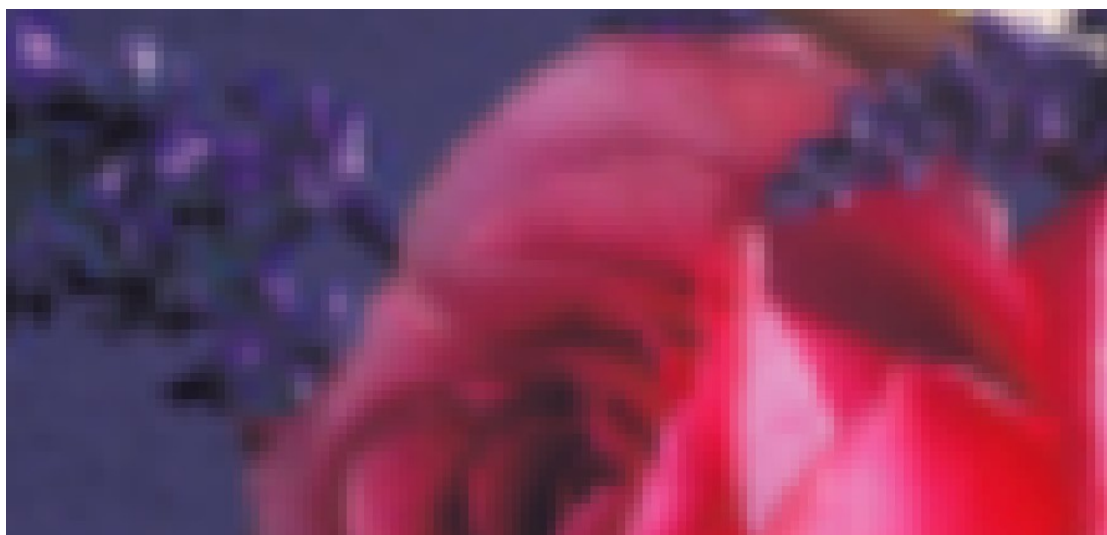
SSIM HR: 0.905451



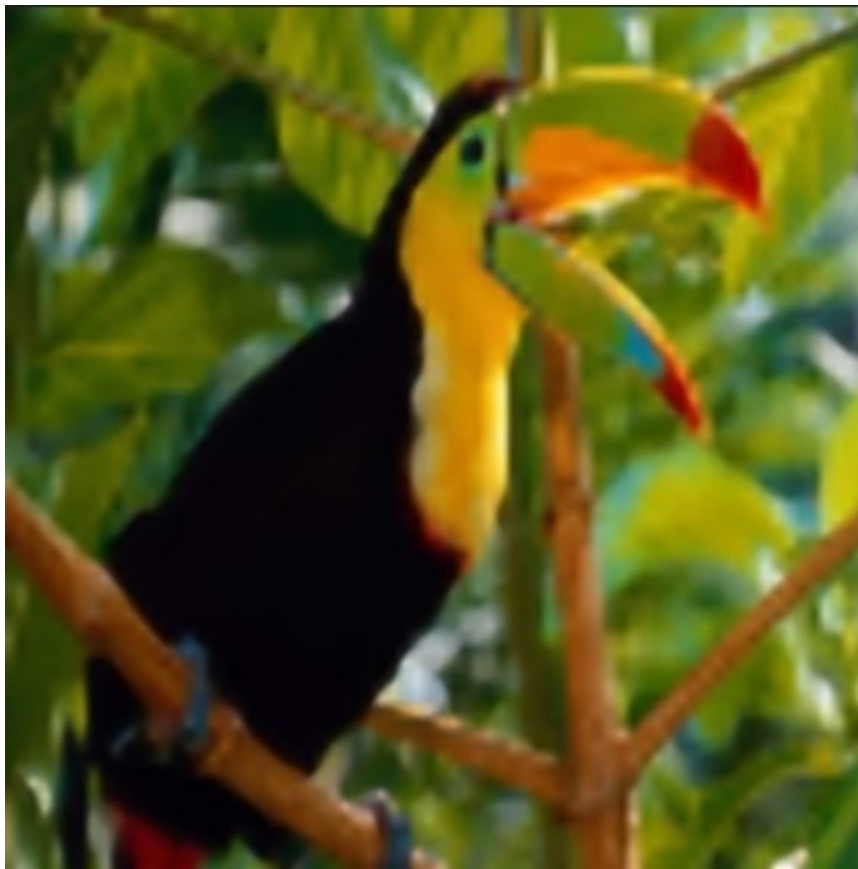
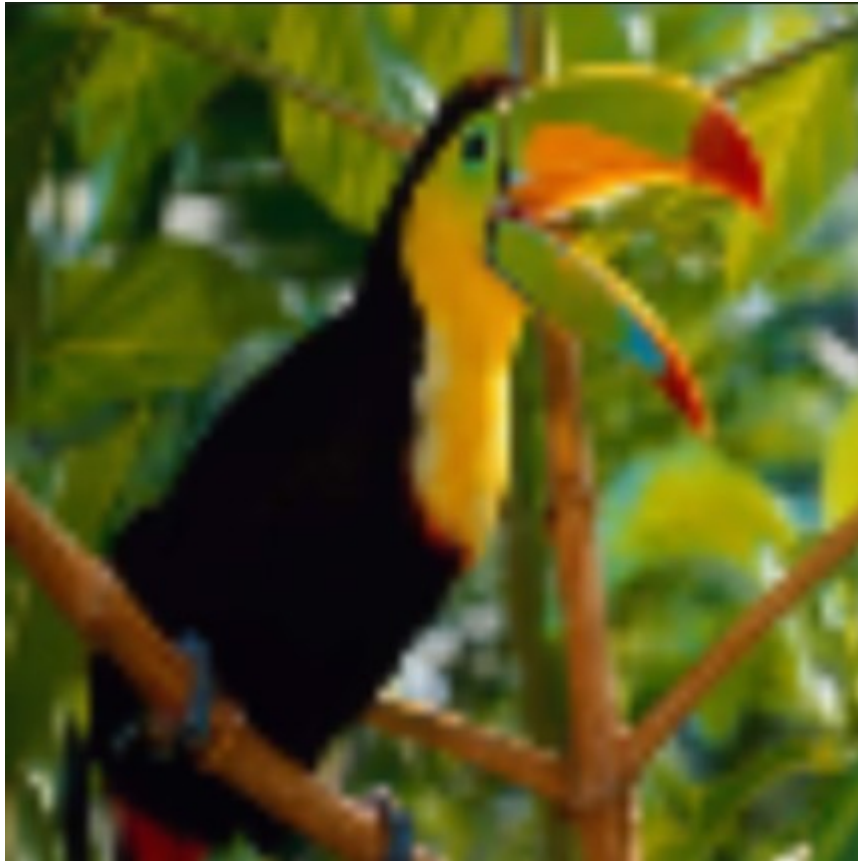
















**Filtri mreže:**

