

```
In [ ]: import math
import random
from os import listdir
from os.path import join

import cv2
import lpips
import matplotlib.pyplot as plt
import numpy as np
import torch
import torch.nn as nn
import torch.optim as optim
import torch.utils.data
import torchmetrics
import torchvision.transforms.functional as TF
from PIL import Image
from torch.utils.data import DataLoader
from torch.utils.data.dataset import Dataset
from torchvision.models.vgg import vgg16
from torchvision.transforms import (CenterCrop, Compose, RandomCrop, Resize,
                                   ToPILImage, ToTensor)
from tqdm.auto import tqdm
```

/home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/tqdm/auto.py:21: TqdmWarning: IPProgress not found. Please update jupyter and ipywidgets. See https://ipywidgets.readthedocs.io/en/stable/user_install.html

```
In [ ]: def initialize_metrics(device):
    return {
        "mse": nn.MSELoss().to(device),
        "psnr": torchmetrics.PeakSignalNoiseRatio(data_range=1.0).to(device),
        "ssim": torchmetrics.StructuralSimilarityIndexMeasure(data_range=1.0).to(device),
        "lpips": lpips.LPIPS(net="alex").to(device)
    }

def compute_metrics(metrics_fns, prediction, target):
    return {
        "mse": metrics_fns["mse"](prediction, target).item(),
        "psnr": metrics_fns["psnr"](prediction, target).item(),
        "ssim": metrics_fns["ssim"](prediction, target).item(),
        "lpips": metrics_fns["lpips"](prediction, target).item()
    }

def process_image_batch(lr_img, hr_img, model, device):
    lr_img = lr_img.unsqueeze(0).to(device)
    hr_img = hr_img.unsqueeze(0).to(device)
    lr_np = lr_img.squeeze(0).cpu().permute(1, 2, 0).numpy()
    bicubic_np = cv2.resize(lr_np, (256, 256), interpolation=cv2.INTER_CUBIC)
    bicubic = torch.tensor(bicubic_np).permute(2, 0, 1).unsqueeze(0).to(device)
    model_output = torch.clip(model(lr_img), 0, 1)
    return lr_img, hr_img, bicubic, model_output

def visualize_image(lr_img, hr_img, bicubic, model_output, ax_row, metrics_bicubic, metrics_model):
    for ax, img, title in zip(ax_row,
                               [lr_img, hr_img, bicubic, model_output],
                               ["Low Res (64x64)", "High Res (256x256)", "Bicubic Interpolation", "Model Output"]):
        img_np = img.squeeze(0).cpu().permute(1, 2, 0).numpy()
        ax.imshow(img_np)
        ax.set_title(title)
        ax.axis("off")

        ax_row[2].text(0.5, -0.15, f"MSE: {metrics_bicubic['mse']:.4f}, PSNR: {metrics_bicubic['psnr']:.2f}\n"
                      f"SSIM: {metrics_bicubic['ssim']:.4f}, LPIPS: {metrics_bicubic['lpips']:.4f}",
                      transform=ax_row[2].transAxes, ha="center", fontsize=10)
        ax_row[3].text(0.5, -0.15, f"MSE: {metrics_model['mse']:.4f}, PSNR: {metrics_model['psnr']:.2f}\n"
                      f"SSIM: {metrics_model['ssim']:.4f}, LPIPS: {metrics_model['lpips']:.4f}",
                      transform=ax_row[3].transAxes, ha="center", fontsize=10)

def evaluate_and_visualize(model, dataset, n=5, device="cuda" if torch.cuda.is_available() else "cpu"):
    model.eval()
    model.to(device)

    metrics_fns = initialize_metrics(device)
    all_metrics = {"bicubic": {"mse": [], "psnr": [], "ssim": [], "lpips": []},
                  "model": {"mse": [], "psnr": [], "ssim": [], "lpips": []}}
    with torch.no_grad():
        for lr_img, hr_img in dataset:
            lr_img, hr_img, bicubic, model_output = process_image_batch(lr_img, hr_img, model, device)
            bicubic_metrics = compute_metrics(metrics_fns, bicubic, hr_img)
            model_metrics = compute_metrics(metrics_fns, model_output, hr_img)
            for metric in all_metrics["bicubic"]:
                all_metrics["bicubic"][metric].append(bicubic_metrics[metric])
            for metric in all_metrics["model"]:
                all_metrics["model"][metric].append(model_metrics[metric])
```

```

        all_metrics["bicubic"][metric].append(bicubic_metrics[metric])
        all_metrics["model"][metric].append(model_metrics[metric])
    avg_metrics = {
        method: {metric: np.mean(values) for metric, values in metrics.items()}
        for method, metrics in all_metrics.items()
    }
    indices = np.random.choice(len(dataset), n, replace=False)
    samples = [dataset[idx] for idx in indices]
    fig, axes = plt.subplots(n, 4, figsize=(20, 5 * n))
    with torch.no_grad():
        for i, (lr_img, hr_img) in enumerate(samples):
            lr_img, hr_img, bicubic, model_output = process_image_batch(lr_img, hr_img, model, device)
            bicubic_metrics = compute_metrics(metrics_fns, bicubic, hr_img)
            model_metrics = compute_metrics(metrics_fns, model_output, hr_img)
            ax_row = axes[i] if n > 1 else axes
            visualize_image(lr_img, hr_img, bicubic, model_output, ax_row, bicubic_metrics, model_metrics)
    print("\nŚrednie metryki dla całego datasetu walidacyjnego:")
    for method in ["bicubic", "model"]:
        print(f"{method.capitalize()} Metrics:")
        print(f"MSE: {avg_metrics[method]['mse']:.4f}, PSNR: {avg_metrics[method]['psnr']:.2f}, "
              f"SSIM: {avg_metrics[method]['ssim']:.4f}, LPIPS: {avg_metrics[method]['lips']:.4f}")
    plt.tight_layout()
    plt.show()

```

```

In [ ]: class Generator(nn.Module):
    def __init__(self, scale_factor):
        super().__init__()
        upsample_block_num = int(math.log(scale_factor, 2))

        self.block1 = nn.Sequential(
            nn.Conv2d(3, 64, kernel_size=9, padding=4), nn.PReLU()
        )

        self.residual_blocks = nn.Sequential(
            *[ResidualBlock(64) for _ in range(5)]
        )

        self.block7 = nn.Sequential(
            nn.Conv2d(64, 64, kernel_size=3, padding=1), nn.BatchNorm2d(64)
        )

        self.block8 = nn.Sequential(
            *[UpsampleBBlock(64, 2) for _ in range(upsample_block_num)],
            nn.Conv2d(64, 3, kernel_size=9, padding=4)
        )

    def forward(self, x):
        block1 = self.block1(x)
        residual = self.residual_blocks(block1)
        block7 = self.block7(residual)
        block8 = self.block8(block1 + block7)
        return (torch.tanh(block8) + 1) / 2

class Discriminator(nn.Module):
    def __init__(self):
        super().__init__()
        self.net = nn.Sequential(
            self._conv_block(3, 64, 3, 1, False),
            self._conv_block(64, 64, 3, 2),
            self._conv_block(64, 128, 3, 1),
            self._conv_block(128, 128, 3, 2),
            self._conv_block(128, 256, 3, 1),
            self._conv_block(256, 256, 3, 2),
            self._conv_block(256, 512, 3, 1),
            self._conv_block(512, 512, 3, 2),
            nn.AdaptiveAvgPool2d(1),
            nn.Conv2d(512, 1024, 1),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Conv2d(1024, 1, 1)
        )

    def _conv_block(self, in_channels, out_channels, kernel_size, stride, batch_norm=True):
        layers = [
            nn.Conv2d(in_channels, out_channels, kernel_size, stride, padding=1),
            nn.LeakyReLU(0.2, inplace=True)
        ]
        if batch_norm:
            layers.insert(1, nn.BatchNorm2d(out_channels))
        return nn.Sequential(*layers)

    def forward(self, x):

```

```

        return torch.sigmoid(self.net(x).view(x.size(0)))

class ResidualBlock(nn.Module):
    def __init__(self, channels):
        super().__init__()
        self.block = nn.Sequential(
            nn.Conv2d(channels, channels, 3, padding=1),
            nn.BatchNorm2d(channels),
            nn.PReLU(),
            nn.Conv2d(channels, channels, 3, padding=1),
            nn.BatchNorm2d(channels)
        )

    def forward(self, x):
        return x + self.block(x)

class UpsampleBLock(nn.Module):
    def __init__(self, in_channels, up_scale):
        super().__init__()
        self.block = nn.Sequential(
            nn.Conv2d(in_channels, in_channels * up_scale ** 2, 3, padding=1),
            nn.PixelShuffle(up_scale),
            nn.PReLU()
        )

    def forward(self, x):
        return self.block(x)

```

```

In [ ]: class GeneratorLoss(nn.Module):
    def __init__(self):
        super().__init__()
        vgg = vgg16(pretrained=True)
        self.loss_network = nn.Sequential(*list(vgg.features)[:31]).eval()
        for param in self.loss_network.parameters():
            param.requires_grad = False
        self.mse_loss = nn.MSELoss()
        self.tv_loss = TVLoss()

    def forward(self, out_labels, out_images, target_images):
        adversarial_loss = torch.mean(1 - out_labels)
        perception_loss = self.mse_loss(self.loss_network(out_images), self.loss_network(target_images))
        image_loss = self.mse_loss(out_images, target_images)
        tv_loss = self.tv_loss(out_images)
        return image_loss + 0.001 * adversarial_loss + 0.006 * perception_loss + 2e-8 * tv_loss

class TVLoss(nn.Module):
    def __init__(self, tv_loss_weight=1):
        super().__init__()
        self.tv_loss_weight = tv_loss_weight

    def forward(self, x):
        batch_size, _, h_x, w_x = x.size()
        h_tv = torch.pow((x[:, :, 1:, :] - x[:, :, :-1, :]), 2).sum()
        w_tv = torch.pow((x[:, :, :, 1:] - x[:, :, :, :-1]), 2).sum()
        return self.tv_loss_weight * 2 * (h_tv / self.tensor_size(x[:, :, 1:, :]) + w_tv / self.tensor_size(x[:, :, :, 1:]))

    @staticmethod
    def tensor_size(t):
        return t.numel() // t.size(0)

```

```

In [ ]: def is_image_file(filename):
    return any(filename.lower().endswith(ext) for ext in ['.png', '.jpg', '.jpeg'])

def calculate_valid_crop_size(crop_size, upscale_factor):
    return crop_size - (crop_size % upscale_factor)

def train_hr_transform(crop_size):
    return Compose([RandomCrop(crop_size), ToTensor()])

def train_lr_transform(crop_size, upscale_factor):
    return Compose([ToPILImage(), Resize(crop_size // upscale_factor, interpolation=Image.BICUBIC), ToTensor()])

class TrainDatasetFromFolder(Dataset):
    def __init__(self, dataset_dir, crop_size, upscale_factor):
        super().__init__()

```

```

        self.image_filenames = [join(dataset_dir, x) for x in listdir(dataset_dir) if is_image_file(x)]
        crop_size = calculate_valid_crop_size(crop_size, upscale_factor)
        self.hr_transform = train_hr_transform(crop_size)
        self.lr_transform = train_lr_transform(crop_size, upscale_factor)
        self.rotation_angles = [0, 90, 180, 270]

    def __getitem__(self, index):
        hr_image = Image.open(self.image_filenames[index]).convert('RGB')
        hr_image = TF.rotate(hr_image, random.choice(self.rotation_angles))
        hr_image = self.hr_transform(hr_image)
        lr_image = self.lr_transform(hr_image)
        return lr_image, hr_image

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

class ValDatasetFromFolder(Dataset):
    def __init__(self, dataset_dir, upscale_factor):
        super().__init__()
        self.upscale_factor = upscale_factor
        self.image_filenames = [join(dataset_dir, x) for x in listdir(dataset_dir) if is_image_file(x)]

    def __getitem__(self, index):
        hr_image = Image.open(self.image_filenames[index])
        crop_size = calculate_valid_crop_size(min(hr_image.size), self.upscale_factor)
        lr_scale = Resize(crop_size // self.upscale_factor, interpolation=Image.BICUBIC)
        hr_image = CenterCrop(crop_size)(hr_image)
        lr_image = lr_scale(hr_image)
        return ToTensor()(lr_image), ToTensor()(hr_image)

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

```

```

In [ ]: def train_gan(config):
    train_set = TrainDatasetFromFolder(config["TRAIN_DATA_PATH"], crop_size=config["CROP_SIZE"], upscale_factor=config["UPSCALE_FACTOR"])
    val_set = ValDatasetFromFolder(config["VAL_DATA_PATH"], upscale_factor=config["UPSCALE_FACTOR"])
    train_loader = DataLoader(dataset=train_set, num_workers=config["NUM_WORKERS"], batch_size=config["BATCH_SIZE"])

    netG = Generator(config["UPSCALE_FACTOR"])
    netD = Discriminator()

    generator_criterion = GeneratorLoss()

    if torch.cuda.is_available():
        netG.cuda()
        netD.cuda()
        generator_criterion.cuda()

    optimizerG = optim.Adam(netG.parameters())
    optimizerD = optim.Adam(netD.parameters())

    for epoch in range(1, config["NUM_EPOCHS"] + 1):
        train_epoch(train_loader, netG, netD, generator_criterion, optimizerG, optimizerD, epoch, config["NUM_EPOCHS"])
        if epoch % 5 == 0:
            netG.eval()
            evaluate_and_visualize(netG, val_set)

def train_epoch(train_loader, netG, netD, generator_criterion, optimizerG, optimizerD, epoch, num_epochs):
    netG.train()
    netD.train()

    running_results = {'batch_sizes': 0, 'd_loss': 0, 'g_loss': 0, 'd_score': 0, 'g_score': 0}

    with tqdm(train_loader, desc=f"Epoch {epoch}/{num_epochs}", leave=True, dynamic_ncols=True) as train_bar:
        for data, target in train_bar:
            batch_size = data.size(0)
            running_results['batch_sizes'] += batch_size

            real_img = target.cuda() if torch.cuda.is_available() else target.float()
            z = data.cuda() if torch.cuda.is_available() else data.float()

            fake_img = netG(z)
            fake_out = netD(fake_img).mean()

            optimizerG.zero_grad()
            g_loss = generator_criterion(fake_out, fake_img, real_img)
            g_loss.backward()
            optimizerG.step()

            real_out = netD(real_img).mean()

```

```

fake_out = netD(fake_img.detach()).mean()
d_loss = 1 - real_out + fake_out

optimizerD.zero_grad()
d_loss.backward()
optimizerD.step()

running_results['g_loss'] += g_loss.item() * batch_size
running_results['d_loss'] += d_loss.item() * batch_size
running_results['d_score'] += real_out.item() * batch_size
running_results['g_score'] += fake_out.item() * batch_size

train_bar.set_postfix({
    "Loss_D": f"{running_results['d_loss'] / running_results['batch_sizes']:.4f}",
    "Loss_G": f"{running_results['g_loss'] / running_results['batch_sizes']:.4f}",
    "D(x)": f"{running_results['d_score'] / running_results['batch_sizes']:.4f}",
    "D(G(z))": f"{running_results['g_score'] / running_results['batch_sizes']:.4f}"
})

```

```
In [ ]: def set_random_seed(seed):
    torch.manual_seed(seed)
    random.seed(seed)
    np.random.seed(seed)

set_random_seed(42)
```

```
In [ ]: config = {
    "CROP_SIZE": 256,
    "UPSCALE_FACTOR": 4,
    "NUM_EPOCHS": 50,
    "BATCH_SIZE": 8,
    "NUM_WORKERS": 4,
    "TRAIN_DATA_PATH": 'data/train/256',
    "VAL_DATA_PATH": 'data/valid/256'
}
train_gan(config)
```

```
/home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/torchvision/models/_utils.py:208: UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the future, please use 'weights' instead.
    warnings.warn(
/home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/torchvision/models/_utils.py:223: UserWarning: Arguments other than a weight enum or `None` for 'weights' are deprecated since 0.13 and may be removed in the future. The current behavior is equivalent to passing `weights=VGG16_Weights.IMAGENET1K_V1`. You can also use `weights=VGG16_Weights.DEFAULT` to get the most up-to-date weights.
    warnings.warn(msg)
Epoch 1/50: 100%|██████████| 100/100 [01:39<00:00,  1.01it/s, Loss_D=0.9910, Loss_G=0.0305, D(x)=0.5296, D(G(z))=0.5205]
Epoch 2/50: 100%|██████████| 100/100 [01:40<00:00,  1.00s/it, Loss_D=0.9990, Loss_G=0.0228, D(x)=0.7896, D(G(z))=0.7886]
Epoch 3/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0001, Loss_G=0.0196, D(x)=0.9995, D(G(z))=0.9996]
Epoch 4/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0183, D(x)=0.9998, D(G(z))=0.9998]
Epoch 5/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0171, D(x)=0.9999, D(G(z))=0.9999]
/home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/torchmetrics/utilities/prints.py:62: FutureWarning: Importing `PeakSignalNoiseRatio` from `torchmetrics` was deprecated and will be removed in 2.0. Import `PeakSignalNoiseRatio` from `torchmetrics.image` instead.
    _future_warning()
/home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/torchmetrics/utilities/prints.py:62: FutureWarning: Importing `StructuralSimilarityIndexMeasure` from `torchmetrics` was deprecated and will be removed in 2.0. Import `StructuralSimilarityIndexMeasure` from `torchmetrics.image` instead.
    _future_warning()
/home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/torchvision/models/_utils.py:223: UserWarning: Arguments other than a weight enum or `None` for 'weights' are deprecated since 0.13 and may be removed in the future. The current behavior is equivalent to passing `weights=AlexNet_Weights.IMAGENET1K_V1`. You can also use `weights=AlexNet_Weights.DEFAULT` to get the most up-to-date weights.
    warnings.warn(msg)
```

```
Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]
Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpipss/weights/v0.1/alex.pth
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.05605055..0.9602562].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.015121151..1.0352201].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.094863586..1.0741736].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.062358424..1.0552871].
```

Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0116, PSNR: 20.22, SSIM: 0.5309, LPIPS: 0.5792

Model Interpolation:

MSE: 0.0130, PSNR: 19.56, SSIM: 0.5074, LPIPS: 0.5353

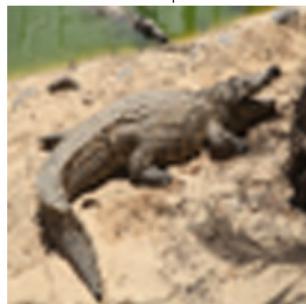
Low Res (64x64)



High Res (256x256)



Bicubic Interpolation



Model Output



Low Res (64x64)



High Res (256x256)



MSE: 0.0117, PSNR: 19.33
SSIM: 0.3785, LPIPS: 0.7658
Bicubic Interpolation

MSE: 0.0124, PSNR: 19.06
SSIM: 0.3805, LPIPS: 0.6979
Model Output



Low Res (64x64)



High Res (256x256)



MSE: 0.0068, PSNR: 21.69
SSIM: 0.6691, LPIPS: 0.4942
Bicubic Interpolation

MSE: 0.0086, PSNR: 20.64
SSIM: 0.6128, LPIPS: 0.4769
Model Output



Low Res (64x64)



High Res (256x256)



MSE: 0.0064, PSNR: 21.97
SSIM: 0.6389, LPIPS: 0.4859
Bicubic Interpolation

MSE: 0.0071, PSNR: 21.48
SSIM: 0.6266, LPIPS: 0.4229
Model Output



Low Res (64x64)



High Res (256x256)



MSE: 0.0129, PSNR: 18.91
SSIM: 0.5311, LPIPS: 0.5462
Bicubic Interpolation

MSE: 0.0138, PSNR: 18.60
SSIM: 0.5178, LPIPS: 0.5000
Model Output



MSE: 0.0089, PSNR: 20.53
SSIM: 0.6102, LPIPS: 0.5411

MSE: 0.0104, PSNR: 19.83
SSIM: 0.5806, LPIPS: 0.4978

```
Epoch 6/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0164, D(x)=0.9999, D(G(z))=0.9999]
Epoch 7/50: 100%|██████████| 100/100 [01:40<00:00,  1.00s/it, Loss_D=1.0000, Loss_G=0.0158, D(x)=0.9999, D(G(z))=0.9999]
Epoch 8/50: 100%|██████████| 100/100 [01:40<00:00,  1.00s/it, Loss_D=1.0000, Loss_G=0.0159, D(x)=0.9999, D(G(z))=0.9999]
Epoch 9/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0158, D(x)=0.9997, D(G(z))=0.9997]
Epoch 10/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0151, D(x)=0.9973, D(G(z))=0.9973]
```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpipss/weights/v0.1/alex.pth

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.0904238..1.0542088].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.03515978..1.0964617].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.014903011..0.99780583].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.017163847..0.98500735].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.064316005..1.069301].
```

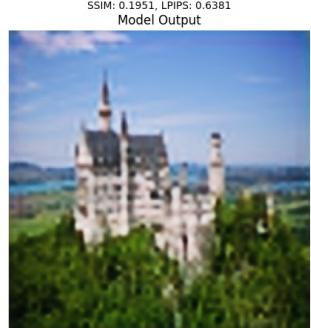
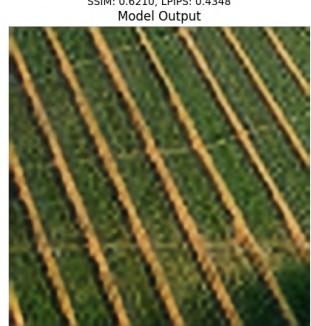
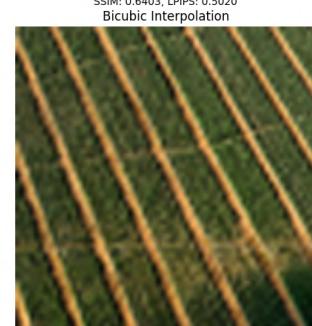
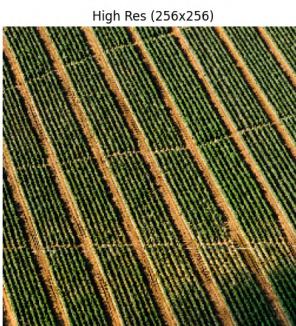
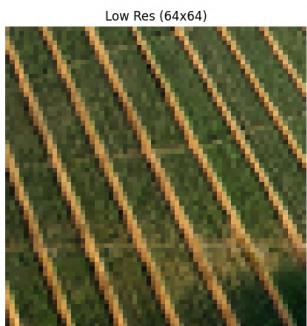
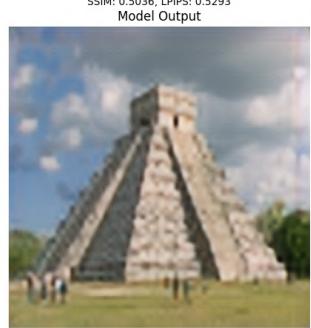
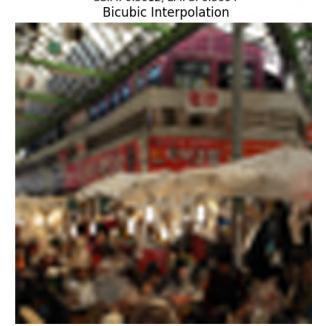
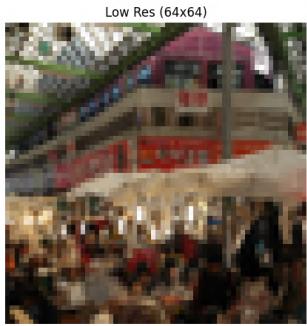
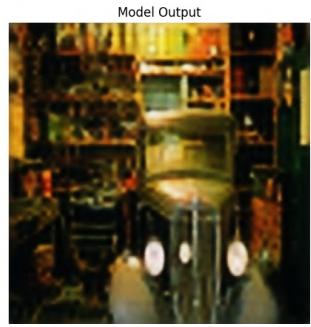
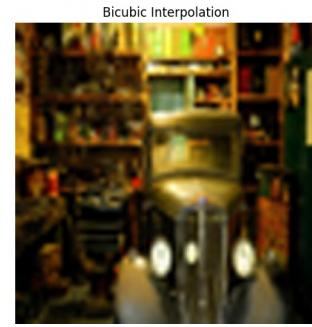
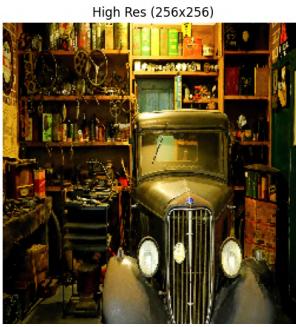
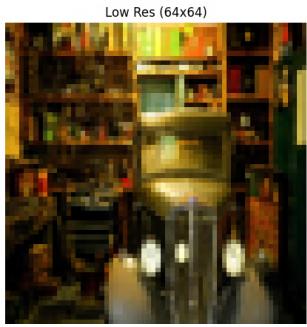
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0116, PSNR: 20.22, SSIM: 0.5309, LPIPS: 0.5792

Model Interpolation:

MSE: 0.0118, PSNR: 20.09, SSIM: 0.5299, LPIPS: 0.4939



```
Epoch 11/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0015, Loss_G=0.0152, D(x)=0.8882, D(G(z))=0.8898]
Epoch 12/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0145, D(x)=0.9998, D(G(z))=0.9998]
Epoch 13/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0142, D(x)=0.9999, D(G(z))=0.9999]
Epoch 14/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0141, D(x)=0.9999, D(G(z))=0.9999]
Epoch 15/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0144, D(x)=0.9999, D(G(z))=0.9999]
```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpips/weights/v0.1/alex.pth

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.014099903..0.991812].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.056648087..1.1030116].

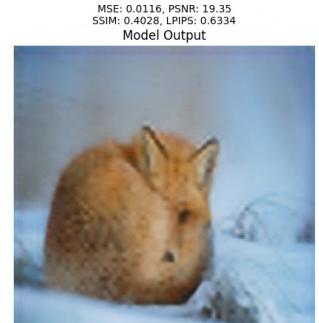
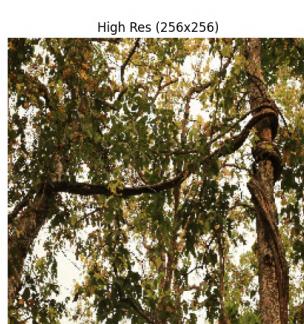
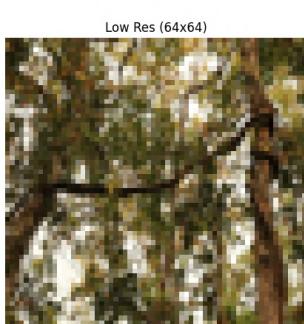
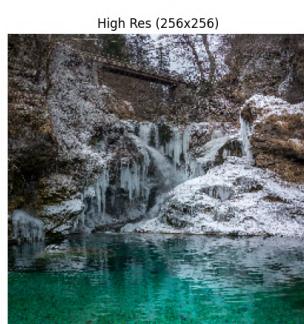
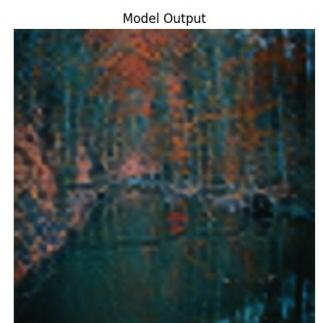
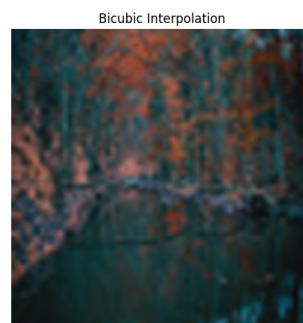
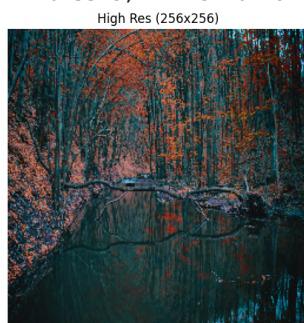
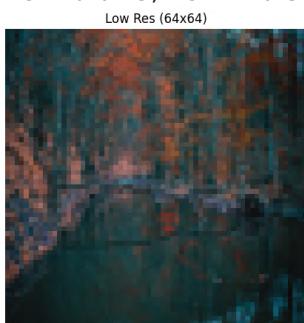
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0116, PSNR: 20.22, SSIM: 0.5309, LPIPS: 0.5792

Model Interpolation:

MSE: 0.0113, PSNR: 20.31, SSIM: 0.5378, LPIPS: 0.4641



MSE: 0.0016, PSNR: 27.98
SSIM: 0.7370, LPIPS: 0.3606

MSE: 0.0018, PSNR: 27.55
SSIM: 0.7166, LPIPS: 0.2836

```
Epoch 16/50: 100%|██████████| 100/100 [01:41<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0142, D(x)=0.9999, D(G(z))=0.9999]
Epoch 17/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0140, D(x)=0.9998, D(G(z))=0.9998]
Epoch 18/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0141, D(x)=0.9998, D(G(z))=0.9998]
Epoch 19/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0139, D(x)=0.9998, D(G(z))=0.9998]
Epoch 20/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0138, D(x)=0.9999, D(G(z))=0.9999]
```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpipss/weights/v0.1/alex.pth

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.009436747..1.023987].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [0.012681359..1.0240862].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.06640732..1.0831865].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.02431945..1.0629778].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.057059444..0.8836831].
```

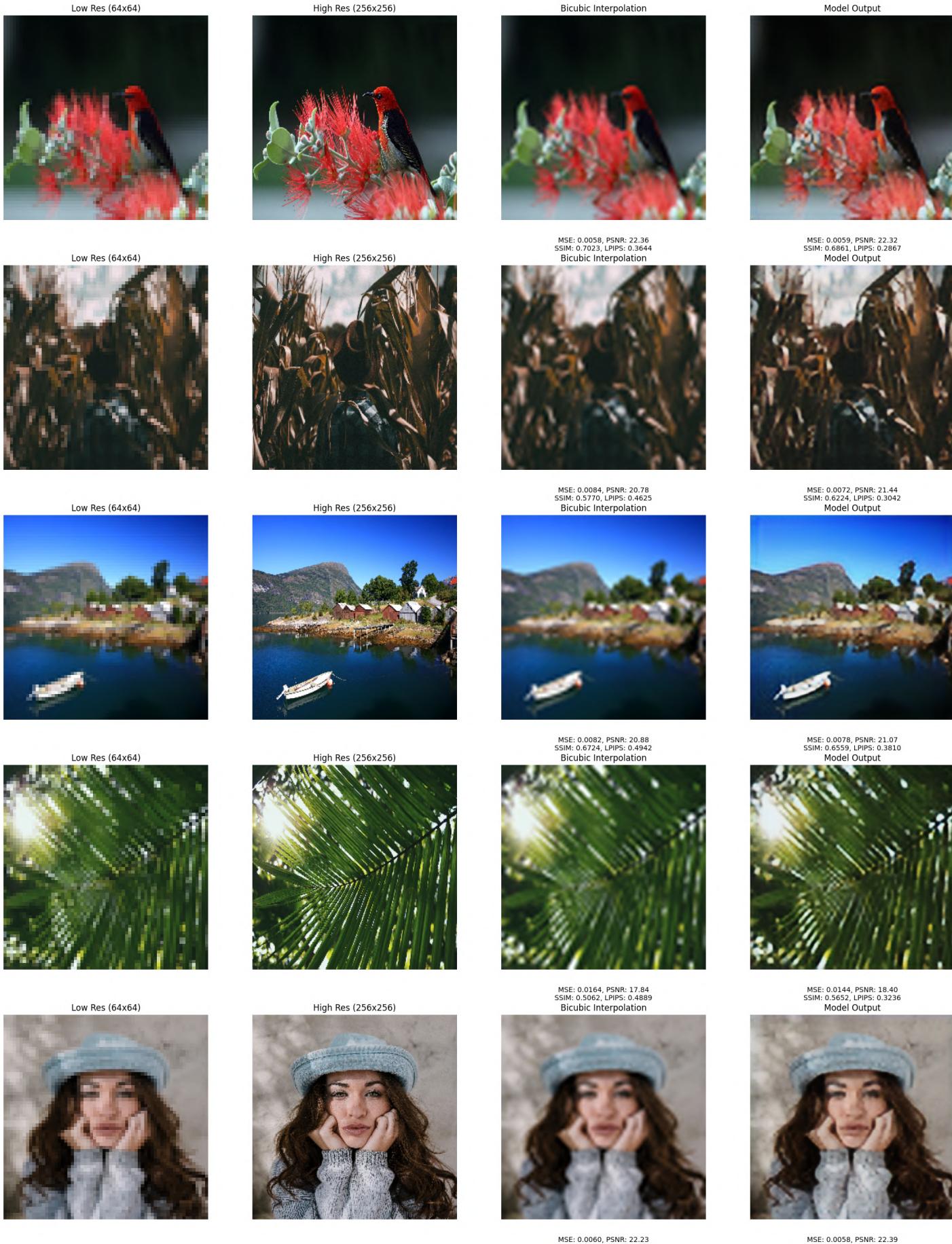
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0116, PSNR: 20.22, SSIM: 0.5309, LPIPS: 0.5792

Model Interpolation:

MSE: 0.0111, PSNR: 20.43, SSIM: 0.5526, LPIPS: 0.4408



```

Epoch 21/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0138, D(x)=0.9997, D(G(z))=0.9997]
Epoch 22/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=0.9999, Loss_G=0.0138, D(x)=0.9964, D(G(z))=0.9962]
Epoch 23/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0004, Loss_G=0.0138, D(x)=0.9587, D(G(z))=0.9591]
Epoch 24/50: 100%|██████████| 100/100 [01:40<00:00,  1.00s/it, Loss_D=1.0000, Loss_G=0.0136, D(x)=1.0000, D(G(z))=1.0000]
Epoch 25/50: 100%|██████████| 100/100 [01:40<00:00,  1.00s/it, Loss_D=1.0000, Loss_G=0.0137, D(x)=1.0000, D(G(z))=1.0000]

```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpips/weights/v0.1/alex.pth

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.009330884..1.0386355].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.08832714..0.73778224].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.07251483..1.0921946].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.011192115..1.0902234].

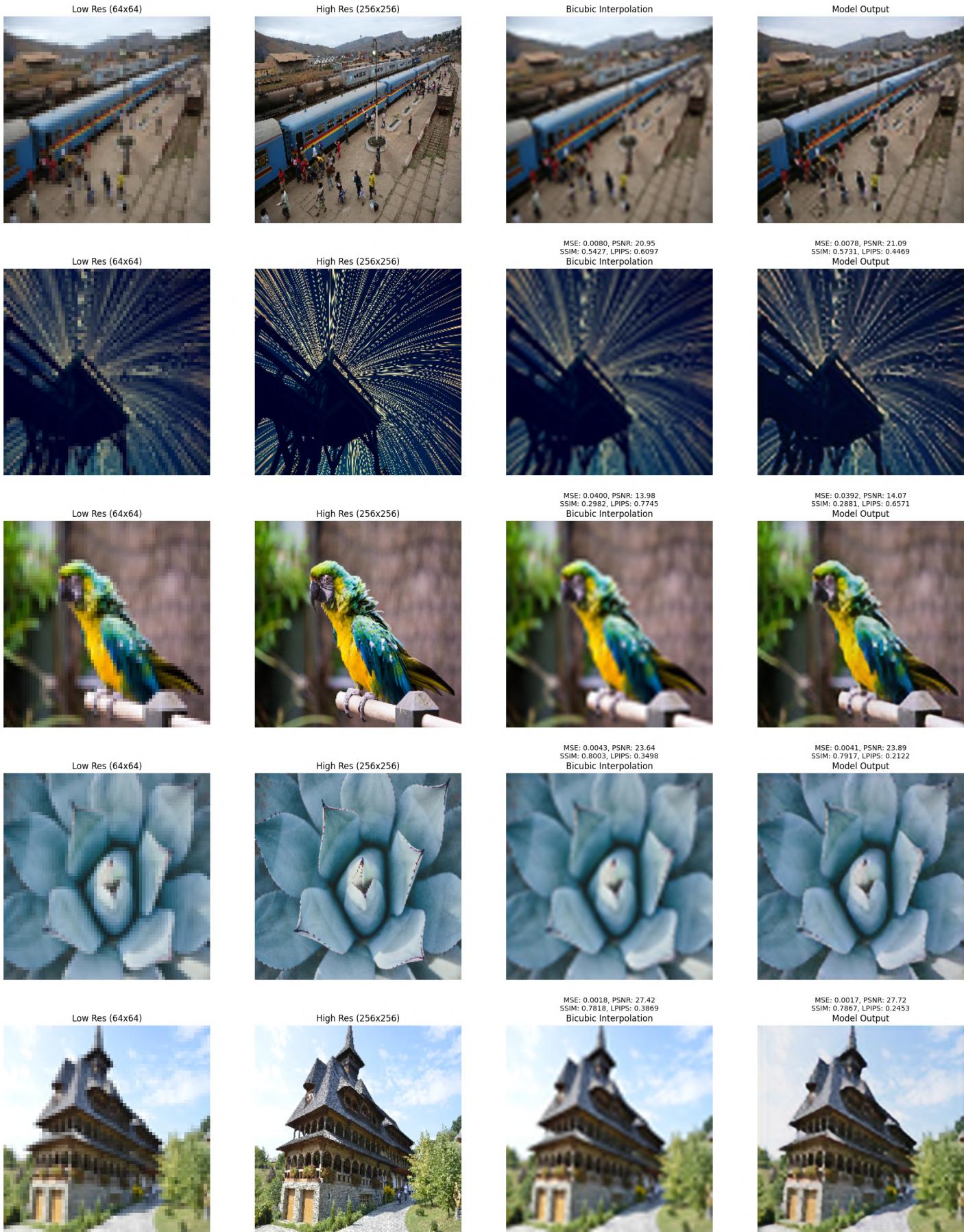
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0116, PSNR: 20.22, SSIM: 0.5309, LPIPS: 0.5792

Model Interpolation:

MSE: 0.0112, PSNR: 20.35, SSIM: 0.5542, LPIPS: 0.4359



MSE: 0.0099, PSNR: 20.02
SSIM: 0.5928, LPIPS: 0.5609

MSE: 0.0104, PSNR: 19.82
SSIM: 0.6191, LPIPS: 0.4063

Bicubic Interpolation

Model Output

MSE: 0.0080, PSNR: 20.95
SSIM: 0.5427, LPIPS: 0.6097
Bicubic Interpolation

MSE: 0.0078, PSNR: 21.09
SSIM: 0.5731, LPIPS: 0.4469
Model Output

MSE: 0.0400, PSNR: 13.98
SSIM: 0.2982, LPIPS: 0.7745
Bicubic Interpolation

MSE: 0.0392, PSNR: 14.07
SSIM: 0.2881, LPIPS: 0.6571
Model Output

MSE: 0.0043, PSNR: 23.64
SSIM: 0.8003, LPIPS: 0.3498
Bicubic Interpolation

MSE: 0.0041, PSNR: 23.89
SSIM: 0.7917, LPIPS: 0.2122
Model Output

MSE: 0.0018, PSNR: 27.42
SSIM: 0.7818, LPIPS: 0.3869
Bicubic Interpolation

MSE: 0.0017, PSNR: 27.72
SSIM: 0.7867, LPIPS: 0.2453
Model Output

```
Epoch 26/50: 100%|██████████| 100/100 [01:40<00:00, 1.00s/it, Loss_D=1.0000, Loss_G=0.0135, D(x)=1.0000, D(G(z))=1.0000]
Epoch 27/50: 100%|██████████| 100/100 [01:40<00:00, 1.00s/it, Loss_D=1.0000, Loss_G=0.0136, D(x)=1.0000, D(G(z))=1.0000]
Epoch 28/50: 100%|██████████| 100/100 [01:40<00:00, 1.00s/it, Loss_D=1.0000, Loss_G=0.0138, D(x)=1.0000, D(G(z))=1.0000]
Epoch 29/50: 100%|██████████| 100/100 [01:40<00:00, 1.00s/it, Loss_D=1.0000, Loss_G=0.0136, D(x)=1.0000, D(G(z))=1.0000]
Epoch 30/50: 100%|██████████| 100/100 [01:40<00:00, 1.00s/it, Loss_D=1.0000, Loss_G=0.0135, D(x)=1.0000, D(G(z))=1.0000]
```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpipss/weights/v0.1/alex.pth

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.062358424..1.0552871].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [0.04523592..1.0471524].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.012436826..0.9334531].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.061894532..1.0366203].

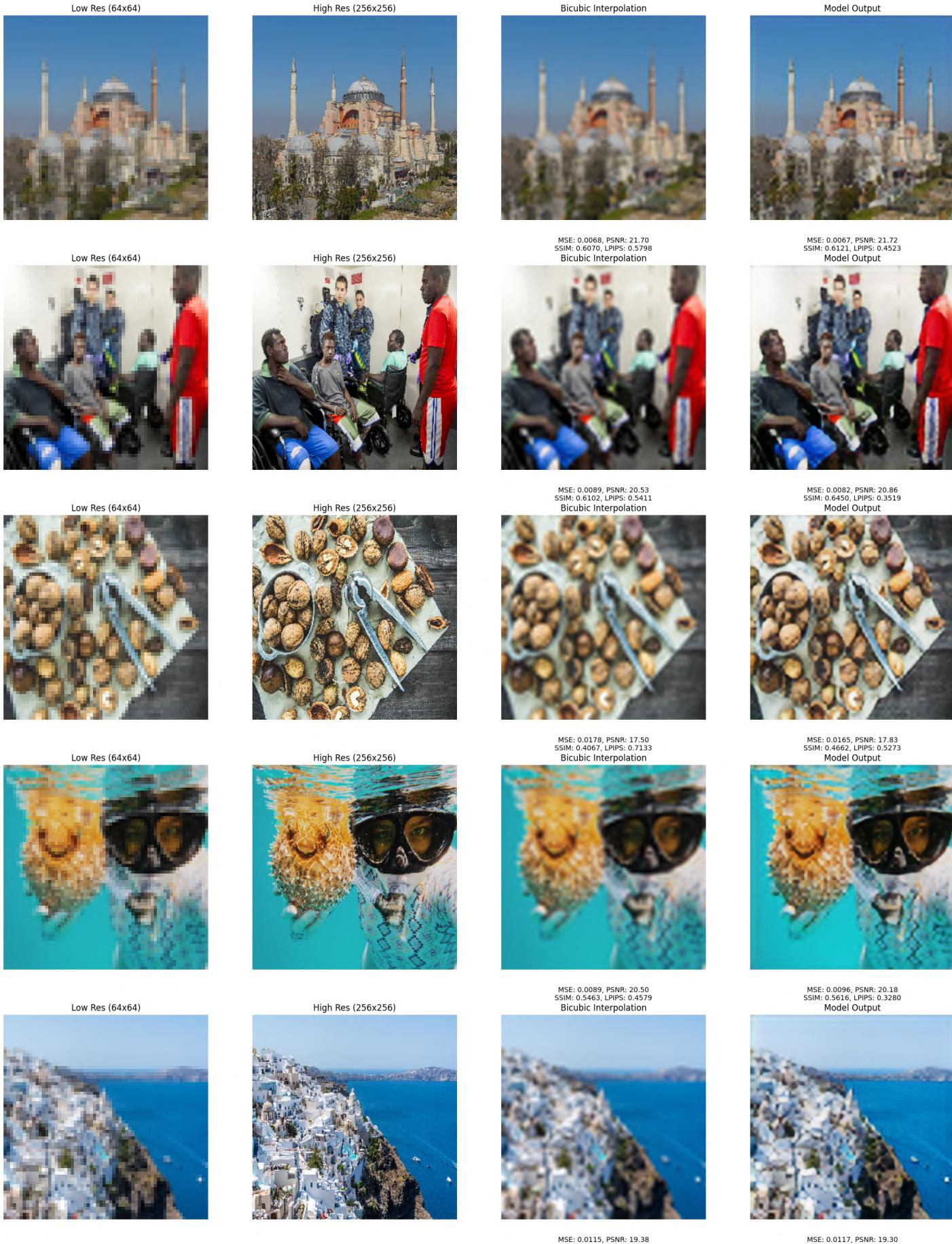
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0116, PSNR: 20.22, SSIM: 0.5309, LPIPS: 0.5792

Model Interpolation:

MSE: 0.0111, PSNR: 20.41, SSIM: 0.5593, LPIPS: 0.4289



```

Epoch 31/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0136, D(x)=1.0000, D(G(z))=1.0000]
Epoch 32/50: 100%|██████████| 100/100 [01:40<00:00,  1.00s/it, Loss_D=1.0000, Loss_G=0.0136, D(x)=1.0000, D(G(z))=1.0000]
Epoch 33/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0134, D(x)=1.0000, D(G(z))=1.0000]
Epoch 34/50: 100%|██████████| 100/100 [01:40<00:00,  1.00s/it, Loss_D=1.0000, Loss_G=0.0134, D(x)=1.0000, D(G(z))=1.0000]
Epoch 35/50: 100%|██████████| 100/100 [01:40<00:00,  1.00s/it, Loss_D=1.0000, Loss_G=0.0135, D(x)=1.0000, D(G(z))=1.0000]

```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpips/weights/v0.1/alex.pth

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.06386585..1.0473173].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.0183345..1.0790533].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.052011218..1.0561833].

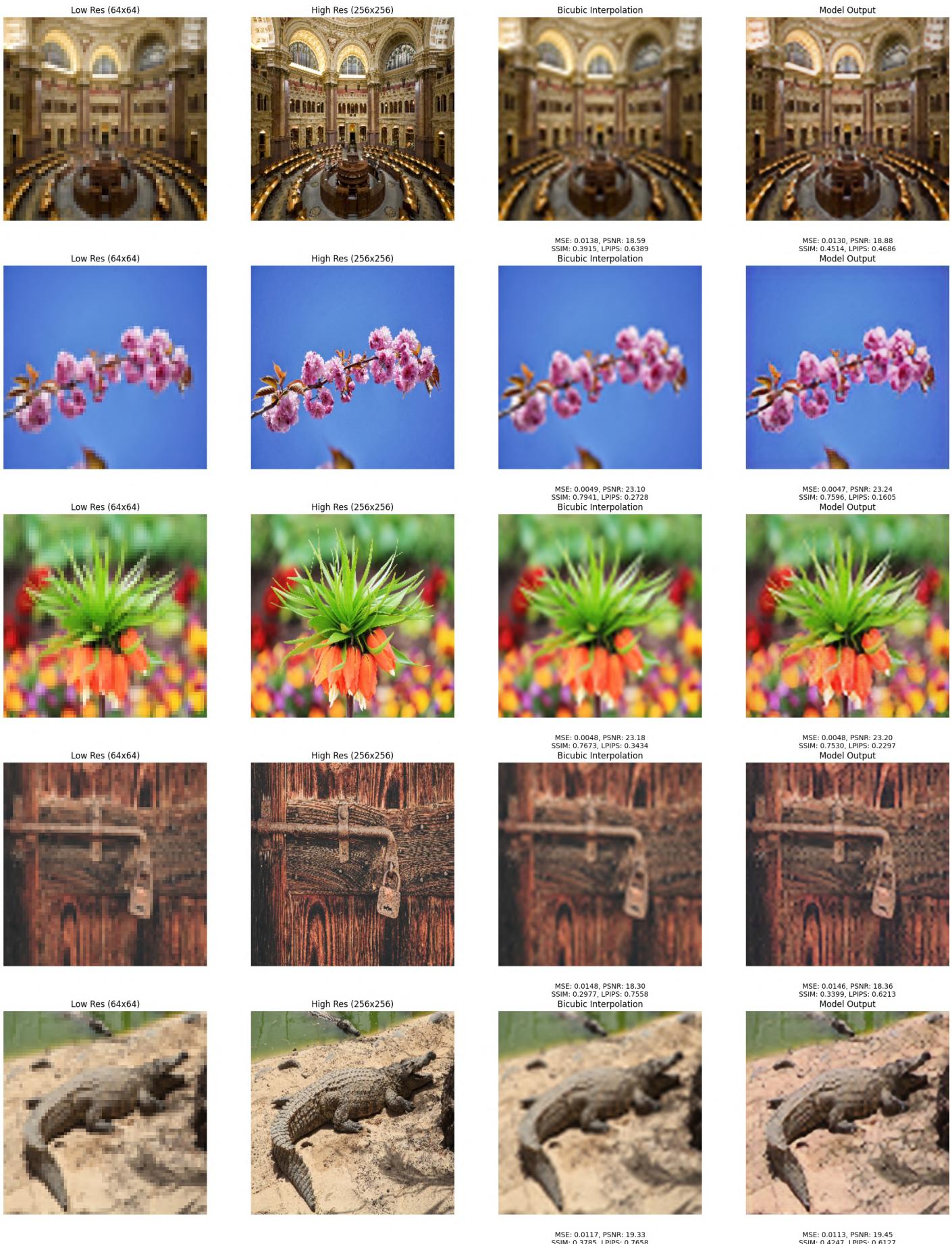
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0116, PSNR: 20.22, SSIM: 0.5309, LPIPS: 0.5792

Model Interpolation:

MSE: 0.0109, PSNR: 20.52, SSIM: 0.5621, LPIPS: 0.4290



```
Epoch 36/50: 100%|██████████| 100/100 [01:40<00:00, 1.01s/it, Loss_D=1.0000, Loss_G=0.0133, D(x)=1.0000, D(G(z))=1.0000]
Epoch 37/50: 100%|██████████| 100/100 [01:40<00:00, 1.01s/it, Loss_D=1.0000, Loss_G=0.0134, D(x)=1.0000, D(G(z))=1.0000]
Epoch 38/50: 100%|██████████| 100/100 [01:40<00:00, 1.01s/it, Loss_D=1.0000, Loss_G=0.0136, D(x)=1.0000, D(G(z))=1.0000]
Epoch 39/50: 100%|██████████| 100/100 [01:40<00:00, 1.01s/it, Loss_D=1.0000, Loss_G=0.0136, D(x)=1.0000, D(G(z))=1.0000]
Epoch 40/50: 100%|██████████| 100/100 [01:40<00:00, 1.01s/it, Loss_D=1.0000, Loss_G=0.0134, D(x)=1.0000, D(G(z))=1.0000]
```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpipss/weights/v0.1/alex.pth

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.0044740913..0.94380367].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.05733994..1.0548062].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.07251483..1.0921946].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.098029755..1.0258217].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.08832714..0.73778224].
```

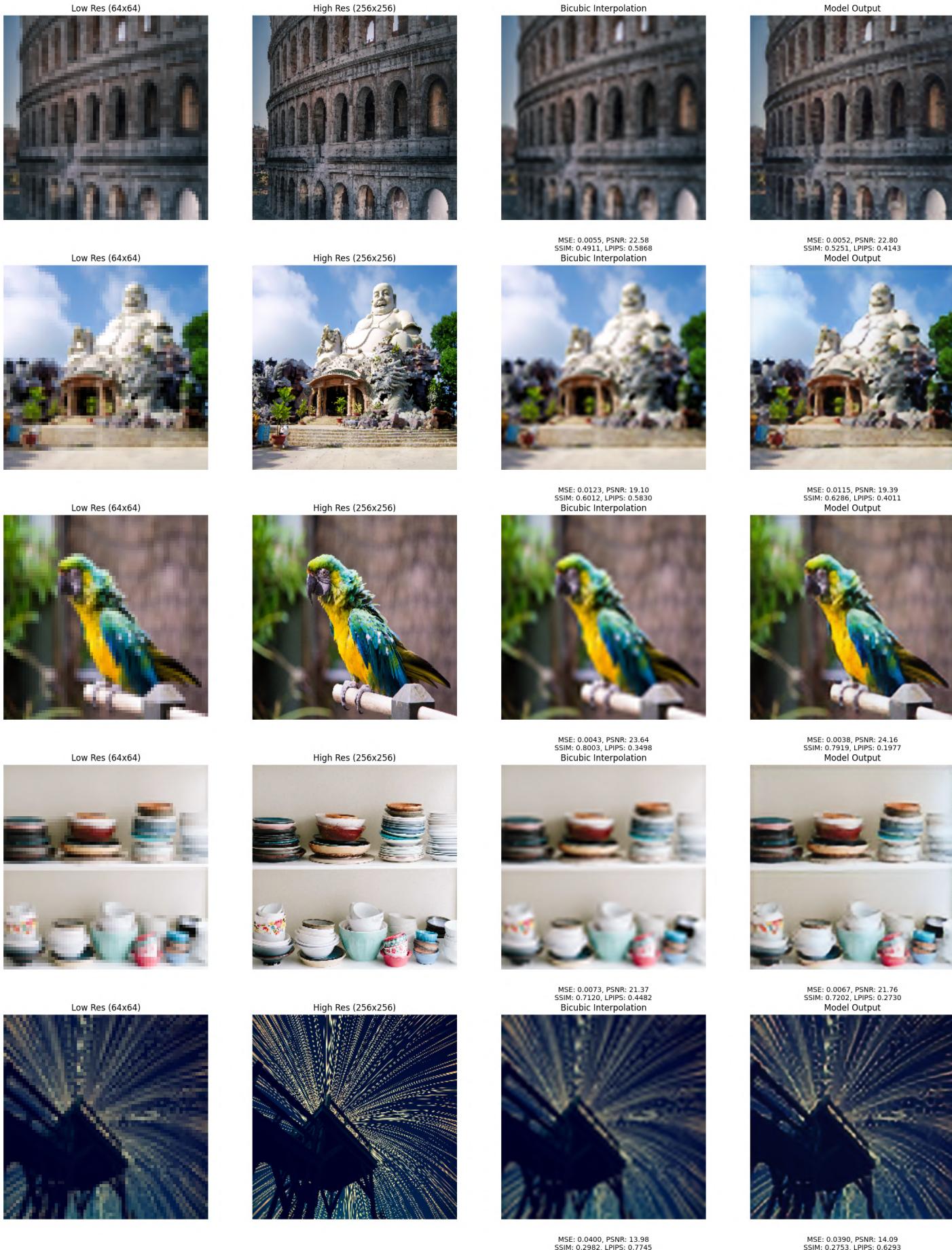
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0116, PSNR: 20.22, SSIM: 0.5309, LPIPS: 0.5792

Model Interpolation:

MSE: 0.0109, PSNR: 20.50, SSIM: 0.5549, LPIPS: 0.4238



```

Epoch 41/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0135, D(x)=1.0000, D(G(z))=1.0000]
Epoch 42/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0132, D(x)=1.0000, D(G(z))=1.0000]
Epoch 43/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0133, D(x)=1.0000, D(G(z))=1.0000]
Epoch 44/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0133, D(x)=1.0000, D(G(z))=1.0000]
Epoch 45/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0133, D(x)=1.0000, D(G(z))=1.0000]

```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpips/weights/v0.1/alex.pth

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.10202327..1.0287253].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.018017035..0.9726162].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.056648087..1.1030116].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [0.047563665..1.0669883].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [0.04452234..1.0108398].

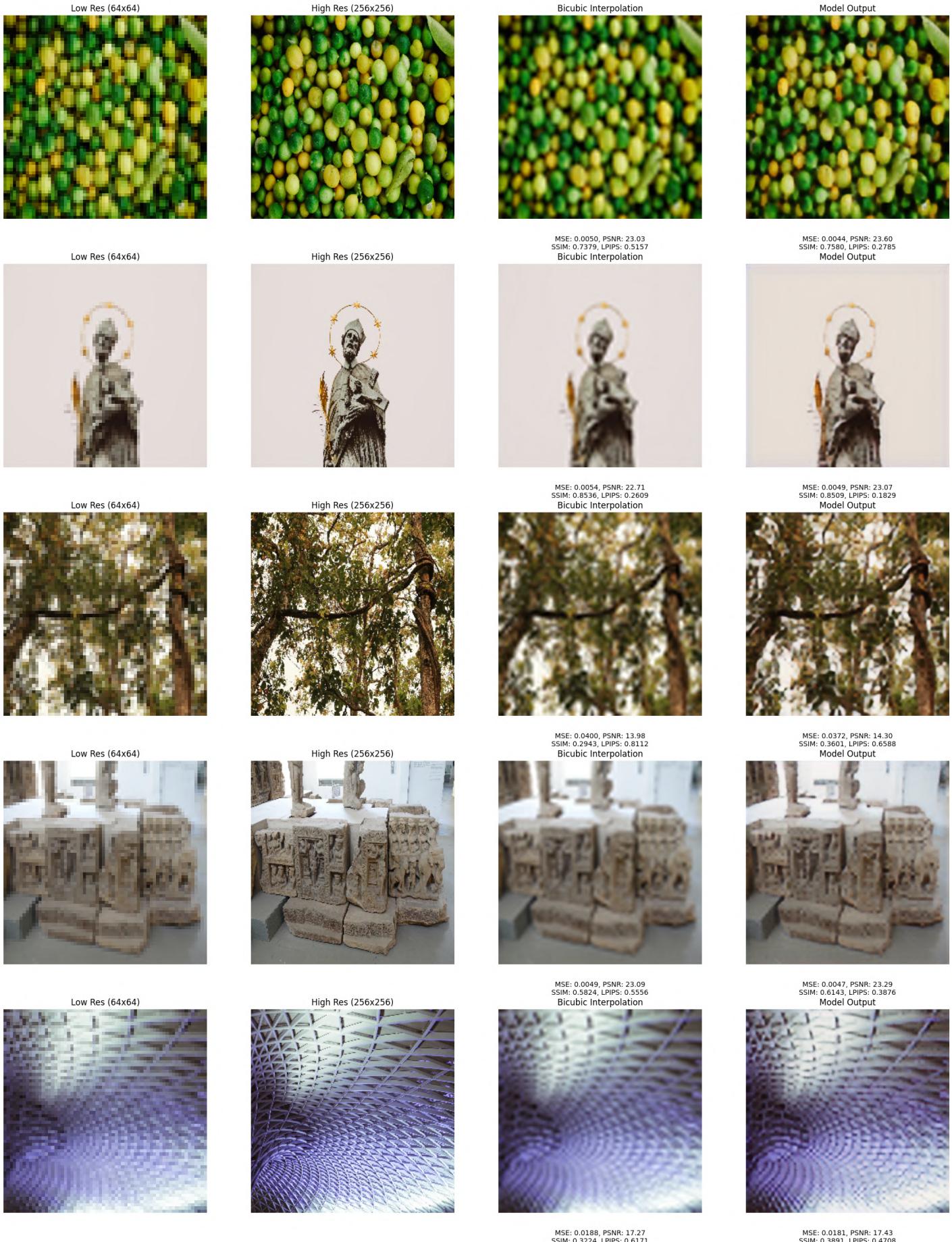
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0116, PSNR: 20.22, SSIM: 0.5309, LPIPS: 0.5792

Model Interpolation:

MSE: 0.0110, PSNR: 20.46, SSIM: 0.5582, LPIPS: 0.4231



```

Epoch 46/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0132, D(x)=1.0000, D(G(z))=1.0000]
Epoch 47/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0133, D(x)=1.0000, D(G(z))=1.0000]
Epoch 48/50: 100%|██████████| 100/100 [01:40<00:00,  1.00s/it, Loss_D=1.0000, Loss_G=0.0133, D(x)=1.0000, D(G(z))=1.0000]
Epoch 49/50: 100%|██████████| 100/100 [01:40<00:00,  1.00s/it, Loss_D=1.0000, Loss_G=0.0135, D(x)=1.0000, D(G(z))=1.0000]
Epoch 50/50: 100%|██████████| 100/100 [01:40<00:00,  1.01s/it, Loss_D=1.0000, Loss_G=0.0132, D(x)=1.0000, D(G(z))=1.0000]

```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpips/weights/v0.1/alex.pth

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.003190836..1.0250804].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [0.006163507..1.0767612].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.11447264..1.0822406].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.07251483..1.0921946].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.064316005..1.069301].

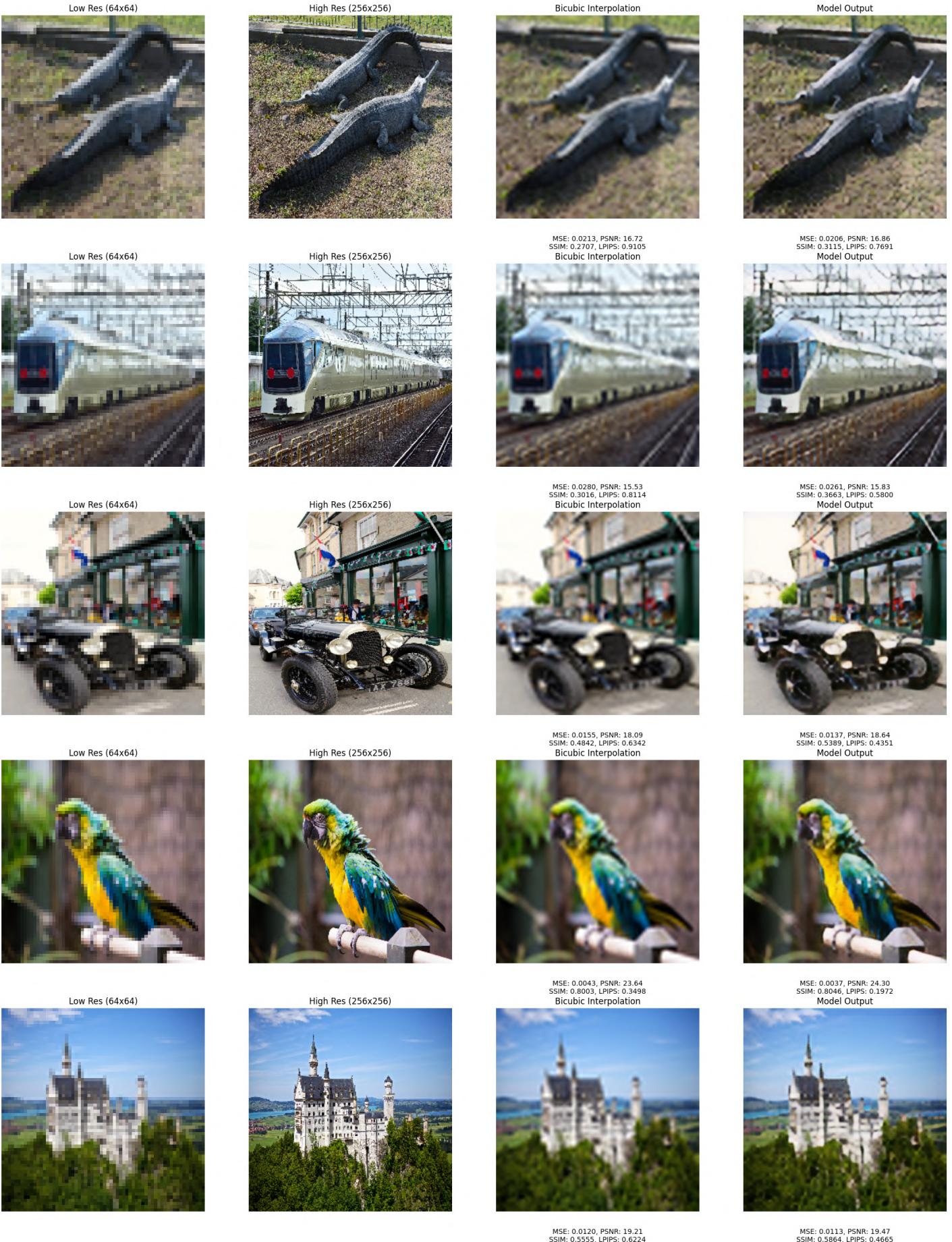
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0116, PSNR: 20.22, SSIM: 0.5309, LPIPS: 0.5792

Model Interpolation:

MSE: 0.0108, PSNR: 20.59, SSIM: 0.5664, LPIPS: 0.4129



```
In [ ]: config = {
    "CROP_SIZE": 256,
    "UPSCALE_FACTOR": 8,
    "NUM_EPOCHS": 50,
    "BATCH_SIZE": 8,
    "NUM_WORKERS": 4,
    "TRAIN_DATA_PATH": 'data/train/256',
    "VAL_DATA_PATH": 'data/valid/256'
}
train_gan(config)
```

```
Epoch 1/50: 100%|██████████| 100/100 [02:01<00:00,  1.21s/it, Loss_D=0.7491, Loss_G=0.0434, D(x)=0.6432, D(G(z))=0.3923]
Epoch 2/50: 100%|██████████| 100/100 [02:14<00:00,  1.34s/it, Loss_D=0.9964, Loss_G=0.0296, D(x)=0.5442, D(G(z))=0.5406]
Epoch 3/50: 100%|██████████| 100/100 [02:14<00:00,  1.34s/it, Loss_D=1.0057, Loss_G=0.0265, D(x)=0.5032, D(G(z))=0.5088]
Epoch 4/50: 100%|██████████| 100/100 [02:13<00:00,  1.34s/it, Loss_D=1.0028, Loss_G=0.0248, D(x)=0.5321, D(G(z))=0.5349]
Epoch 5/50: 100%|██████████| 100/100 [02:14<00:00,  1.34s/it, Loss_D=0.9927, Loss_G=0.0241, D(x)=0.6824, D(G(z))=0.6750]
```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpipss/weights/v0.1/alex.pth

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.015323715..1.1130307].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.09601426..1.0602615].

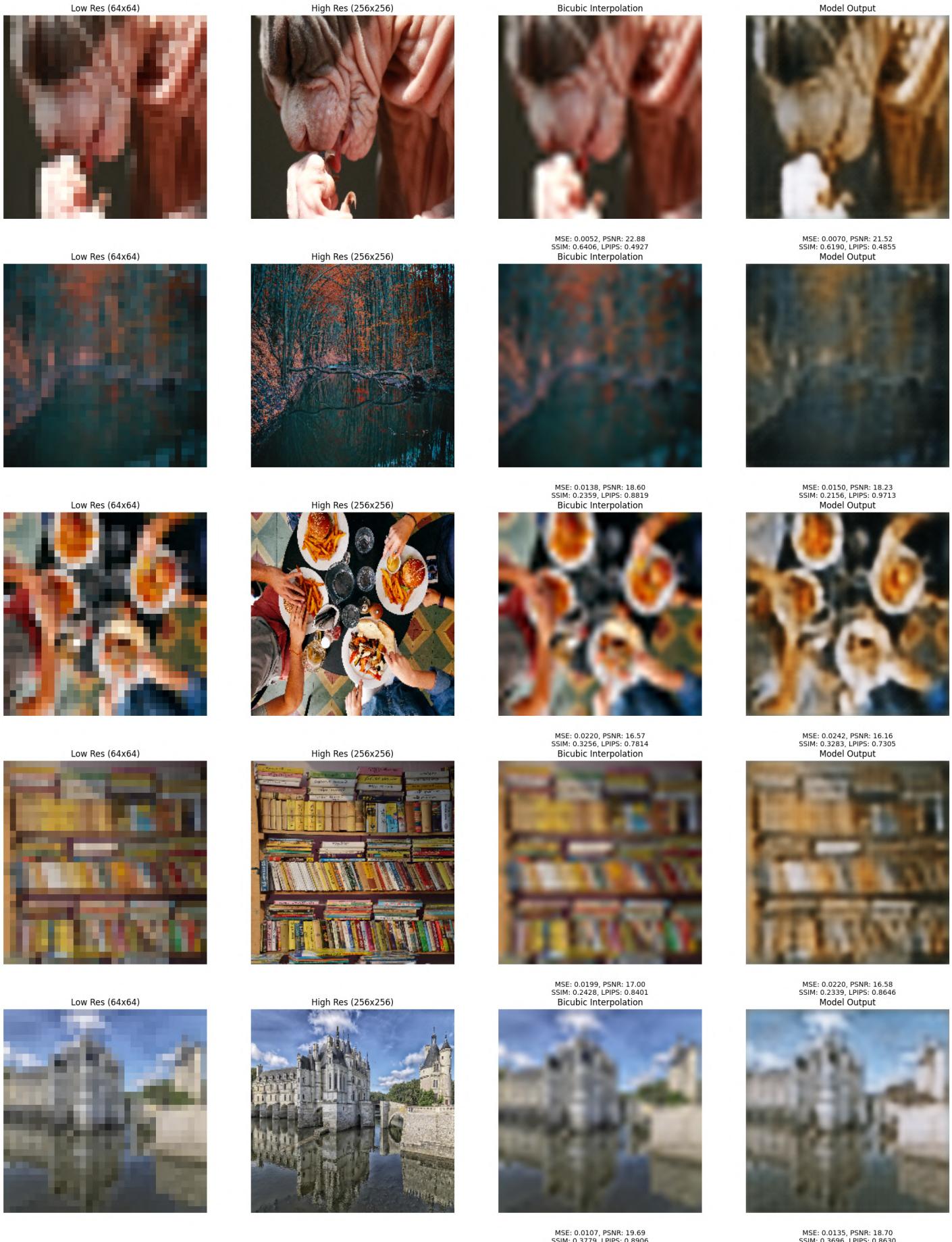
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0167, PSNR: 18.48, SSIM: 0.3862, LPIPS: 0.7725

Model Interpolation:

MSE: 0.0198, PSNR: 17.57, SSIM: 0.3630, LPIPS: 0.7661



```

Epoch 6/50: 100%|██████████| 100/100 [02:14<00:00,  1.34s/it, Loss_D=1.0059, Loss_G=0.0226, D(x)=0.6845, D(G(z))=0.6905]
Epoch 7/50: 100%|██████████| 100/100 [02:11<00:00,  1.32s/it, Loss_D=1.0033, Loss_G=0.0215, D(x)=0.8834, D(G(z))=0.8866]
Epoch 8/50: 100%|██████████| 100/100 [02:09<00:00,  1.30s/it, Loss_D=1.0000, Loss_G=0.0210, D(x)=1.0000, D(G(z))=1.0000]
Epoch 9/50: 100%|██████████| 100/100 [02:09<00:00,  1.30s/it, Loss_D=1.0000, Loss_G=0.0208, D(x)=1.0000, D(G(z))=1.0000]
Epoch 10/50: 100%|██████████| 100/100 [02:09<00:00,  1.30s/it, Loss_D=1.0000, Loss_G=0.0203, D(x)=1.0000, D(G(z))=1.0000]

```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpips/weights/v0.1/alex.pth

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.00067171734..0.814208].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.02003925..0.7593234].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.0048239306..0.82278925].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.0022200004..0.9978649].

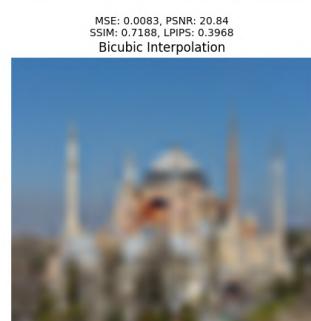
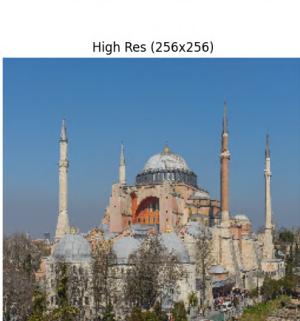
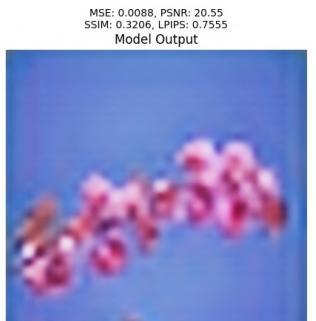
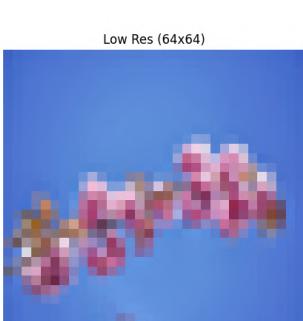
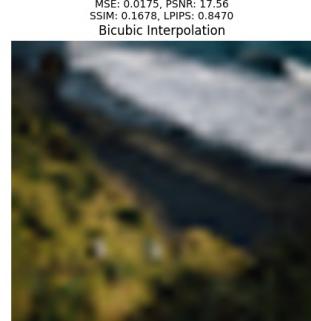
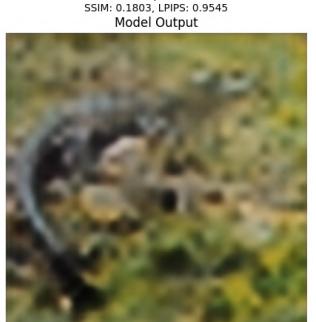
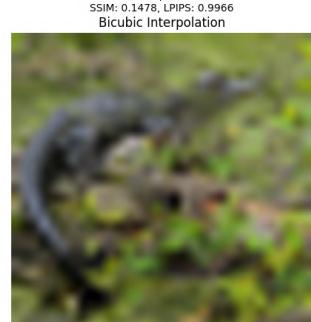
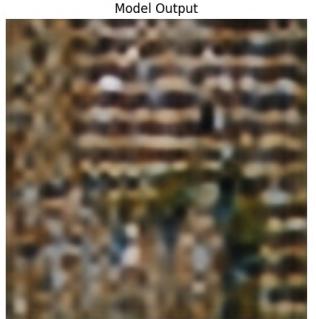
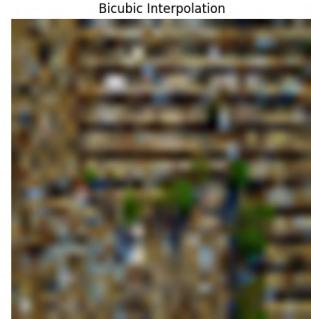
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0167, PSNR: 18.48, SSIM: 0.3862, LPIPS: 0.7725

Model Interpolation:

MSE: 0.0172, PSNR: 18.26, SSIM: 0.3837, LPIPS: 0.7149



MSE: 0.0088, PSNR: 20.58
SSIM: 0.5242, LPIPS: 0.7543

MSE: 0.0096, PSNR: 20.19
SSIM: 0.5008, LPIPS: 0.7425

```
Epoch 11/50: 100%|██████████| 100/100 [01:54<00:00,  1.14s/it, Loss_D=1.0000, Loss_G=0.0202, D(x)=1.0000, D(G(z))=1.0000]
Epoch 12/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0201, D(x)=1.0000, D(G(z))=1.0000]
Epoch 13/50: 100%|██████████| 100/100 [01:37<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0199, D(x)=1.0000, D(G(z))=1.0000]
Epoch 14/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0198, D(x)=1.0000, D(G(z))=1.0000]
Epoch 15/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0199, D(x)=1.0000, D(G(z))=1.0000]
```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpipss/weights/v0.1/alex.pth

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [0.008250146..1.0380698].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.015682442..0.60326964].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.059872262..1.085201].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.0022200004..0.9978649].

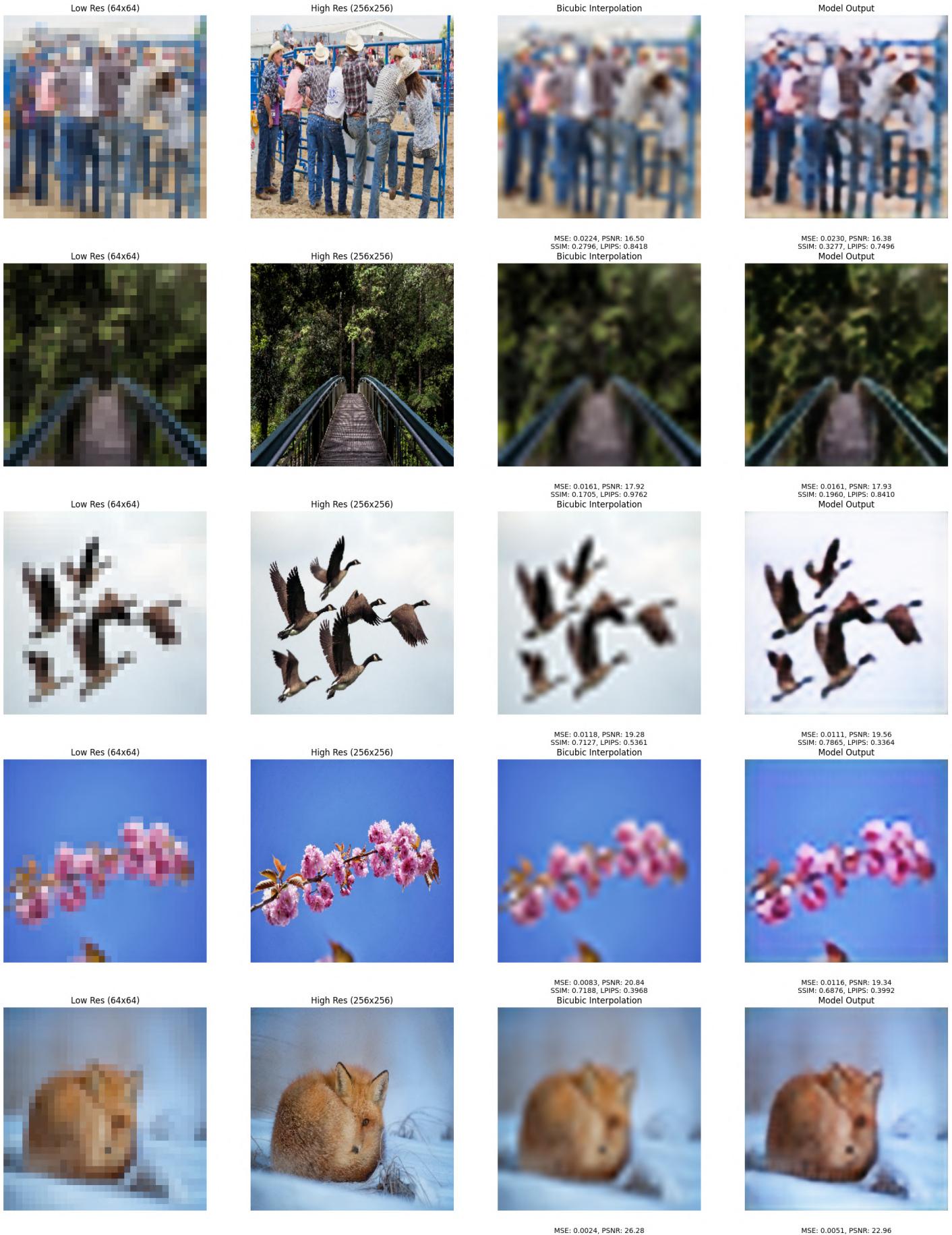
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0167, PSNR: 18.48, SSIM: 0.3862, LPIPS: 0.7725

Model Interpolation:

MSE: 0.0177, PSNR: 18.09, SSIM: 0.3980, LPIPS: 0.6749



```

Epoch 16/50: 100%|██████████| 100/100 [01:37<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0197, D(x)=1.0000, D(G(z))=1.0000]
Epoch 17/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0194, D(x)=1.0000, D(G(z))=1.0000]
Epoch 18/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0193, D(x)=1.0000, D(G(z))=1.0000]
Epoch 19/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0194, D(x)=1.0000, D(G(z))=1.0000]
Epoch 20/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0194, D(x)=1.0000, D(G(z))=1.0000]

```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpips/weights/v0.1/alex.pth

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.008499262..1.0299215].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.047149587..1.0767672].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.03605902..0.5276773].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.034565173..0.8567265].

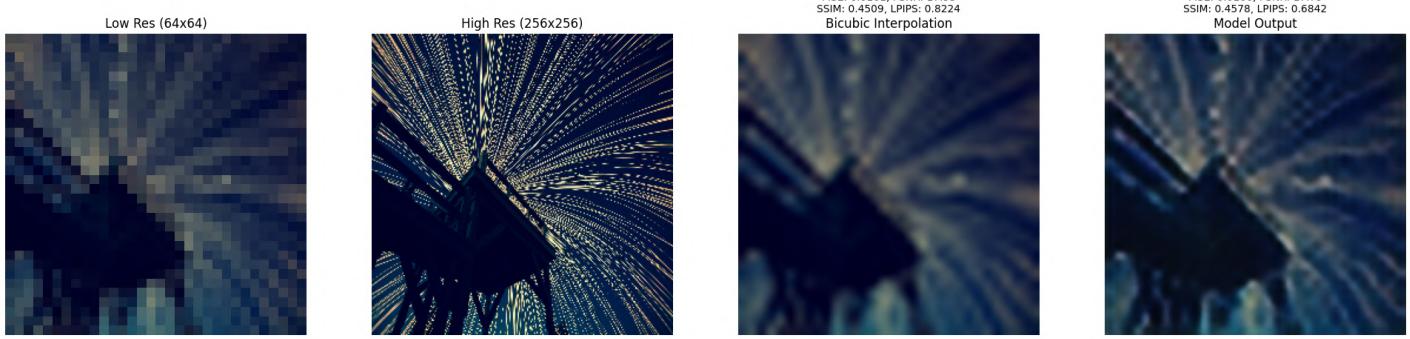
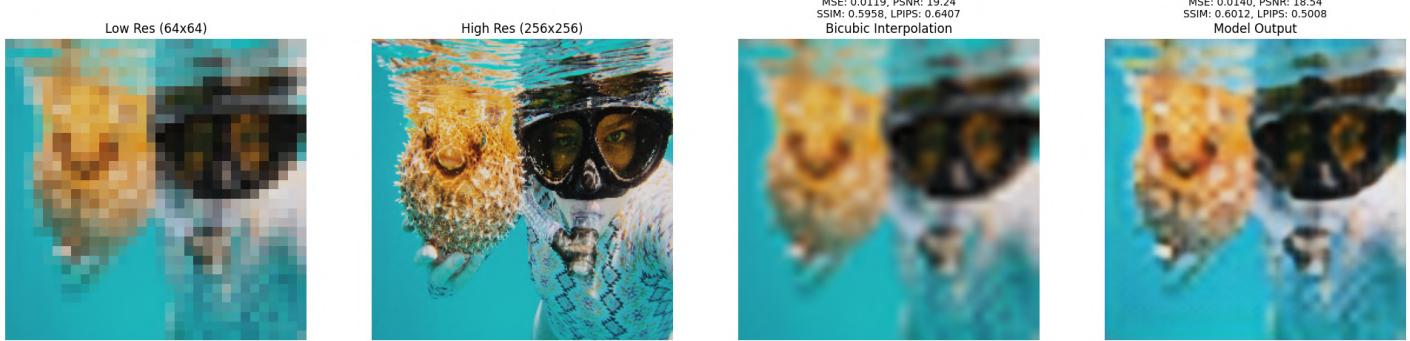
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0167, PSNR: 18.48, SSIM: 0.3862, LPIPS: 0.7725

Model Interpolation:

MSE: 0.0168, PSNR: 18.38, SSIM: 0.4005, LPIPS: 0.6626



MSE: 0.0072, PSNR: 21.40
SSIM: 0.6155, LPIPS: 0.6591

MSE: 0.0076, PSNR: 21.22
SSIM: 0.6162, LPIPS: 0.5430

```
Epoch 21/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0197, D(x)=1.0000, D(G(z))=1.0000]
Epoch 22/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0194, D(x)=1.0000, D(G(z))=1.0000]
Epoch 23/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0192, D(x)=1.0000, D(G(z))=1.0000]
Epoch 24/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0191, D(x)=1.0000, D(G(z))=1.0000]
Epoch 25/50: 100%|██████████| 100/100 [01:36<00:00,  1.04it/s, Loss_D=1.0000, Loss_G=0.0191, D(x)=1.0000, D(G(z))=1.0000]
```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpipss/weights/v0.1/alex.pth

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.043015268..1.0422579].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.035520412..0.9850342].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [0.00411315..1.0347494].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.05115088..1.0385303].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [0.043094967..1.0710195].
```

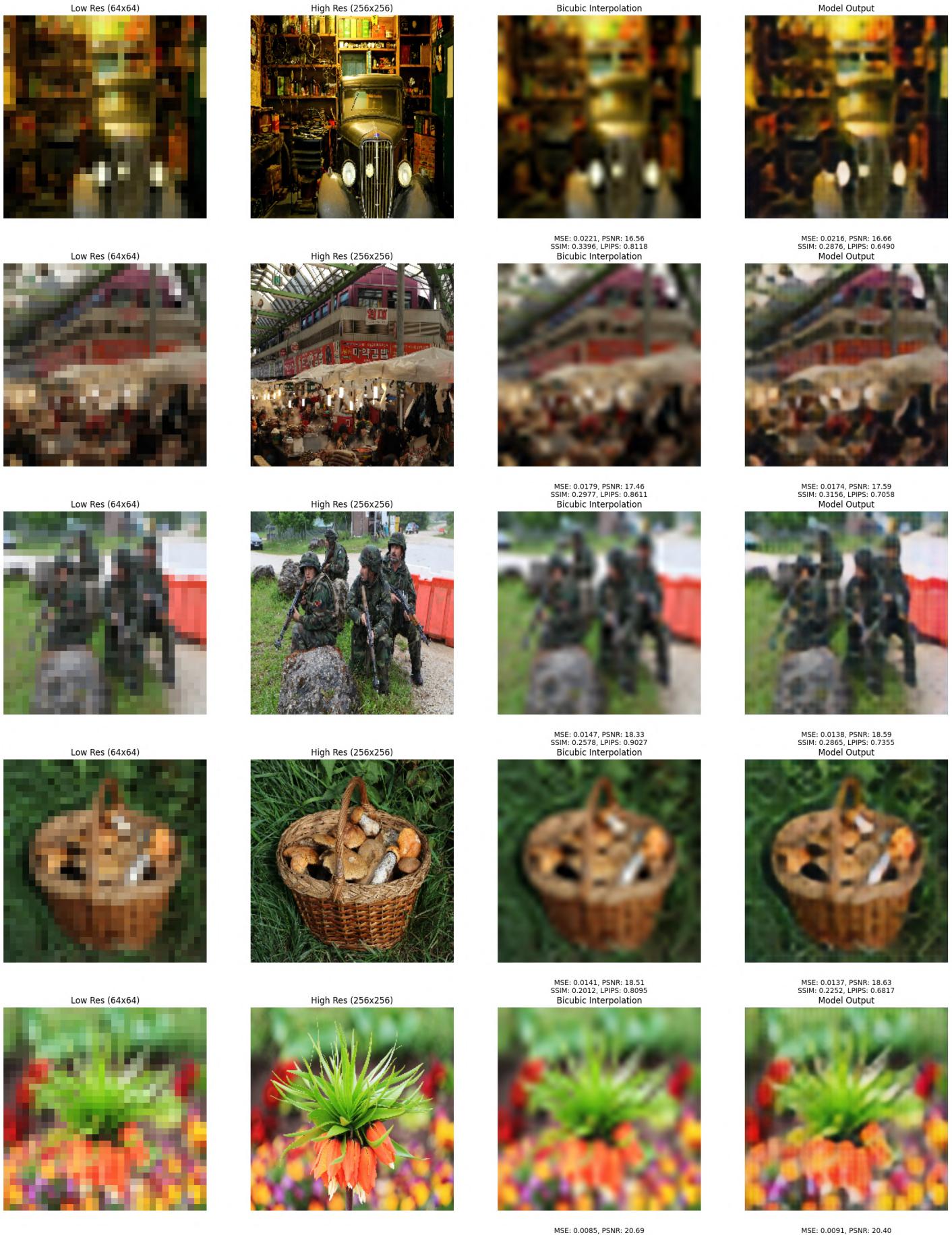
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0167, PSNR: 18.48, SSIM: 0.3862, LPIPS: 0.7725

Model Interpolation:

MSE: 0.0163, PSNR: 18.56, SSIM: 0.3898, LPIPS: 0.6378



```

Epoch 26/50: 100%|██████████| 100/100 [01:36<00:00,  1.04it/s, Loss_D=1.0000, Loss_G=0.0190, D(x)=1.0000, D(G(z))=1.0000]
Epoch 27/50: 100%|██████████| 100/100 [01:36<00:00,  1.04it/s, Loss_D=1.0000, Loss_G=0.0191, D(x)=1.0000, D(G(z))=1.0000]
Epoch 28/50: 100%|██████████| 100/100 [01:36<00:00,  1.04it/s, Loss_D=1.0000, Loss_G=0.0190, D(x)=1.0000, D(G(z))=1.0000]
Epoch 29/50: 100%|██████████| 100/100 [01:36<00:00,  1.04it/s, Loss_D=1.0000, Loss_G=0.0189, D(x)=1.0000, D(G(z))=1.0000]
Epoch 30/50: 100%|██████████| 100/100 [01:36<00:00,  1.04it/s, Loss_D=1.0000, Loss_G=0.0190, D(x)=1.0000, D(G(z))=1.0000]

```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpips/weights/v0.1/alex.pth

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.059872262..1.085201].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.08628853..1.078829].
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.031504136..1.0196886].

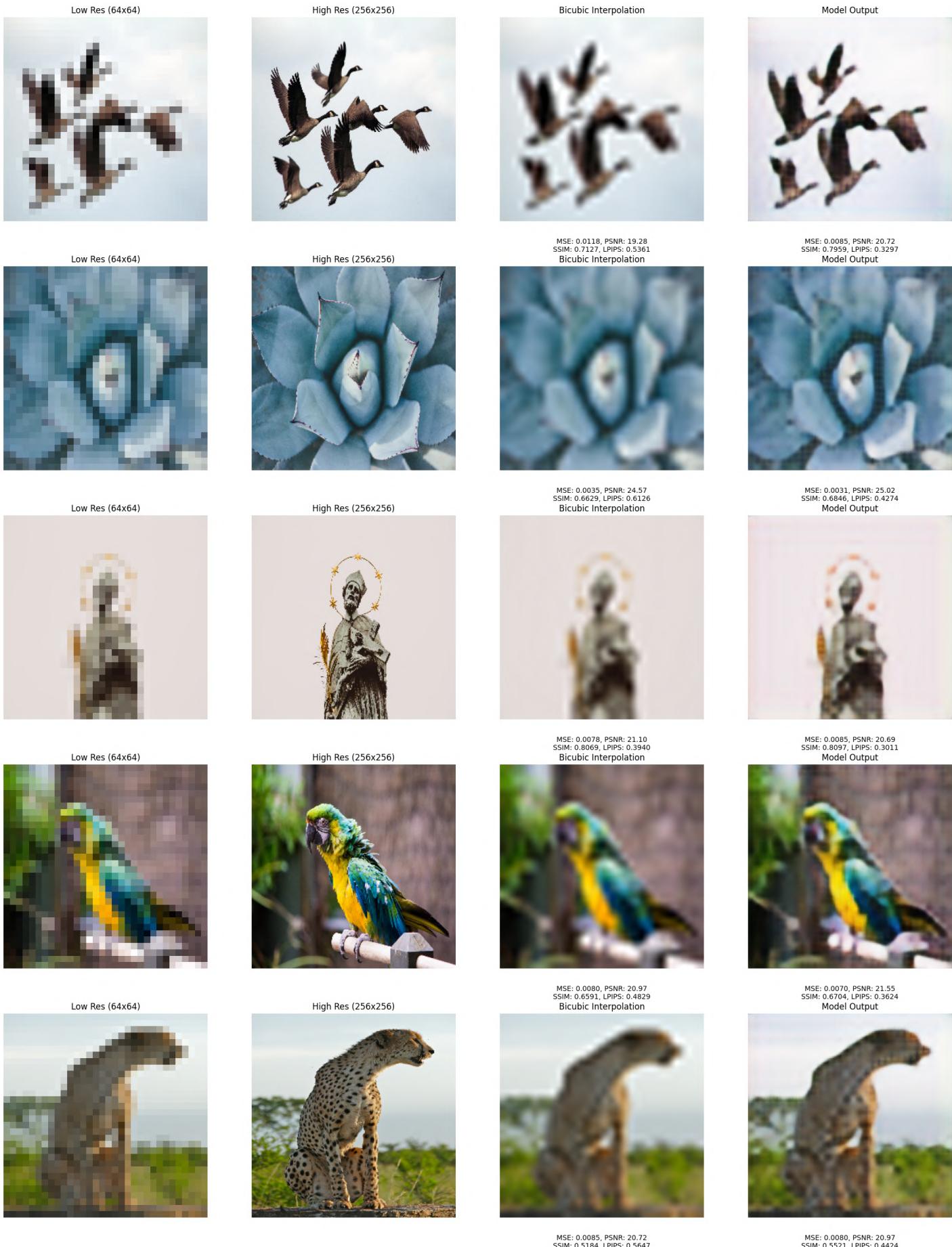
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0167, PSNR: 18.48, SSIM: 0.3862, LPIPS: 0.7725

Model Interpolation:

MSE: 0.0160, PSNR: 18.66, SSIM: 0.4019, LPIPS: 0.6354



```
Epoch 31/50: 100%|██████████| 100/100 [01:36<00:00,  1.04it/s, Loss_D=1.0000, Loss_G=0.0189, D(x)=1.0000, D(G(z))=1.0000]
Epoch 32/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0189, D(x)=1.0000, D(G(z))=1.0000]
Epoch 33/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0188, D(x)=1.0000, D(G(z))=1.0000]
Epoch 34/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0188, D(x)=1.0000, D(G(z))=1.0000]
Epoch 35/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0187, D(x)=1.0000, D(G(z))=1.0000]
```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpipss/weights/v0.1/alex.pth

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.015682442..0.60326964].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.0022200004..0.9978649].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [0.0025252693..1.0876286].
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.047149587..1.0767672].
```

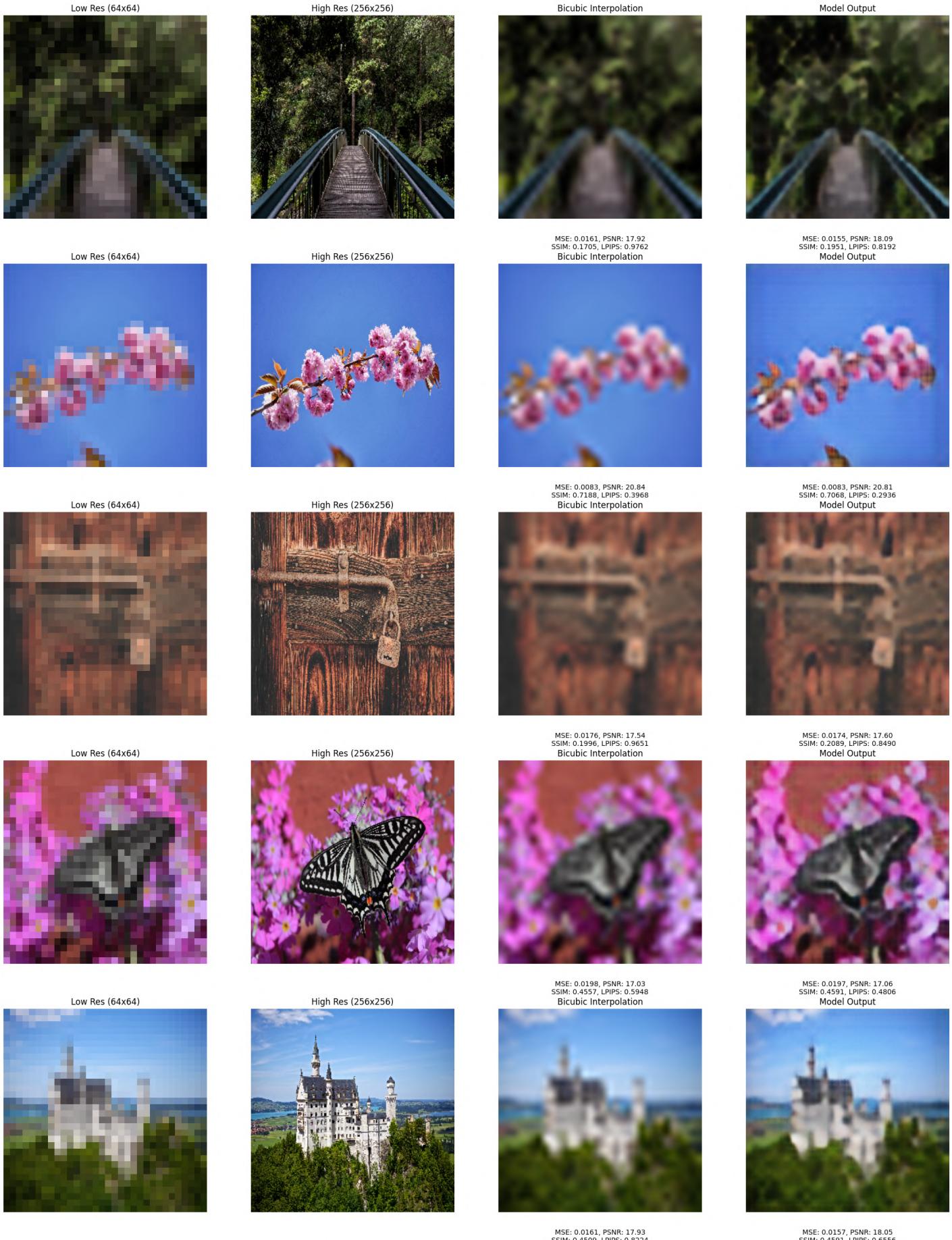
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0167, PSNR: 18.48, SSIM: 0.3862, LPIPS: 0.7725

Model Interpolation:

MSE: 0.0158, PSNR: 18.75, SSIM: 0.4046, LPIPS: 0.6279



```

Epoch 36/50: 100%|██████████| 100/100 [01:35<00:00,  1.04it/s, Loss_D=1.0000, Loss_G=0.0187, D(x)=1.0000, D(G(z))=1.0000]
Epoch 37/50: 100%|██████████| 100/100 [01:36<00:00,  1.04it/s, Loss_D=1.0000, Loss_G=0.0188, D(x)=1.0000, D(G(z))=1.0000]
Epoch 38/50: 100%|██████████| 100/100 [01:36<00:00,  1.04it/s, Loss_D=1.0000, Loss_G=0.0188, D(x)=1.0000, D(G(z))=1.0000]
Epoch 39/50: 100%|██████████| 100/100 [01:36<00:00,  1.04it/s, Loss_D=1.0000, Loss_G=0.0187, D(x)=1.0000, D(G(z))=1.0000]
Epoch 40/50: 100%|██████████| 100/100 [01:36<00:00,  1.04it/s, Loss_D=1.0000, Loss_G=0.0187, D(x)=1.0000, D(G(z))=1.0000]

```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpips/weights/v0.1/alex.pth

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.02234462..1.0837507].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.0048239306..0.82278925].

Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0167, PSNR: 18.48, SSIM: 0.3862, LPIPS: 0.7725

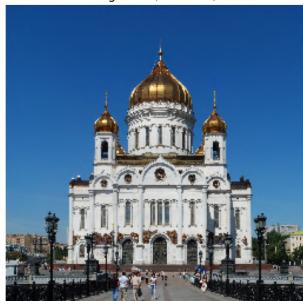
Model Interpolation:

MSE: 0.0157, PSNR: 18.76, SSIM: 0.4036, LPIPS: 0.6226

Low Res (64x64)



High Res (256x256)



Bicubic Interpolation



Model Output



Low Res (64x64)



High Res (256x256)



MSE: 0.0167, PSNR: 17.78
SSIM: 0.5566, LPIPS: 0.6347
Bicubic Interpolation



MSE: 0.0155, PSNR: 18.09
SSIM: 0.5671, LPIPS: 0.4749
Model Output



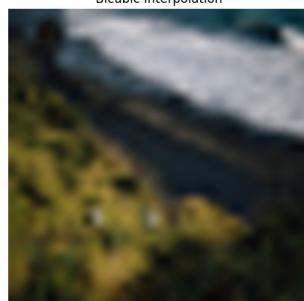
Low Res (64x64)



High Res (256x256)



MSE: 0.0255, PSNR: 15.93
SSIM: 0.1999, LPIPS: 0.9691
Bicubic Interpolation



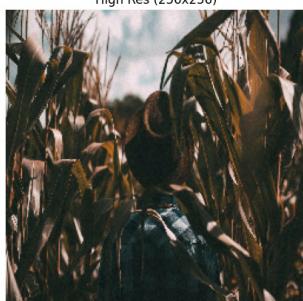
MSE: 0.0244, PSNR: 16.12
SSIM: 0.2292, LPIPS: 0.8287
Model Output



Low Res (64x64)



High Res (256x256)



MSE: 0.0082, PSNR: 20.84
SSIM: 0.3201, LPIPS: 0.8250
Bicubic Interpolation



MSE: 0.0080, PSNR: 20.96
SSIM: 0.3329, LPIPS: 0.6879
Model Output



Low Res (64x64)



High Res (256x256)



MSE: 0.0138, PSNR: 18.60
SSIM: 0.4146, LPIPS: 0.6993
Bicubic Interpolation



MSE: 0.0125, PSNR: 19.04
SSIM: 0.4445, LPIPS: 0.5507
Model Output



MSE: 0.0084, PSNR: 20.73
SSIM: 0.4547, LPIPS: 0.8152

MSE: 0.0084, PSNR: 20.78
SSIM: 0.4626, LPIPS: 0.6911

```
Epoch 41/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0191, D(x)=1.0000, D(G(z))=1.0000]
Epoch 42/50: 100%|██████████| 100/100 [01:36<00:00,  1.04it/s, Loss_D=1.0000, Loss_G=0.0189, D(x)=1.0000, D(G(z))=1.0000]
Epoch 43/50: 100%|██████████| 100/100 [01:36<00:00,  1.03it/s, Loss_D=1.0000, Loss_G=0.0188, D(x)=1.0000, D(G(z))=1.0000]
Epoch 44/50: 100%|██████████| 100/100 [01:36<00:00,  1.04it/s, Loss_D=1.0000, Loss_G=0.0187, D(x)=1.0000, D(G(z))=1.0000]
Epoch 45/50: 100%|██████████| 100/100 [01:36<00:00,  1.04it/s, Loss_D=1.0000, Loss_G=0.0187, D(x)=1.0000, D(G(z))=1.0000]
```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpipss/weights/v0.1/alex.pth

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [0.08339862..1.0268553].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.04190735..0.91447693].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.09772341..0.9871141].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.036058616..1.0352252].

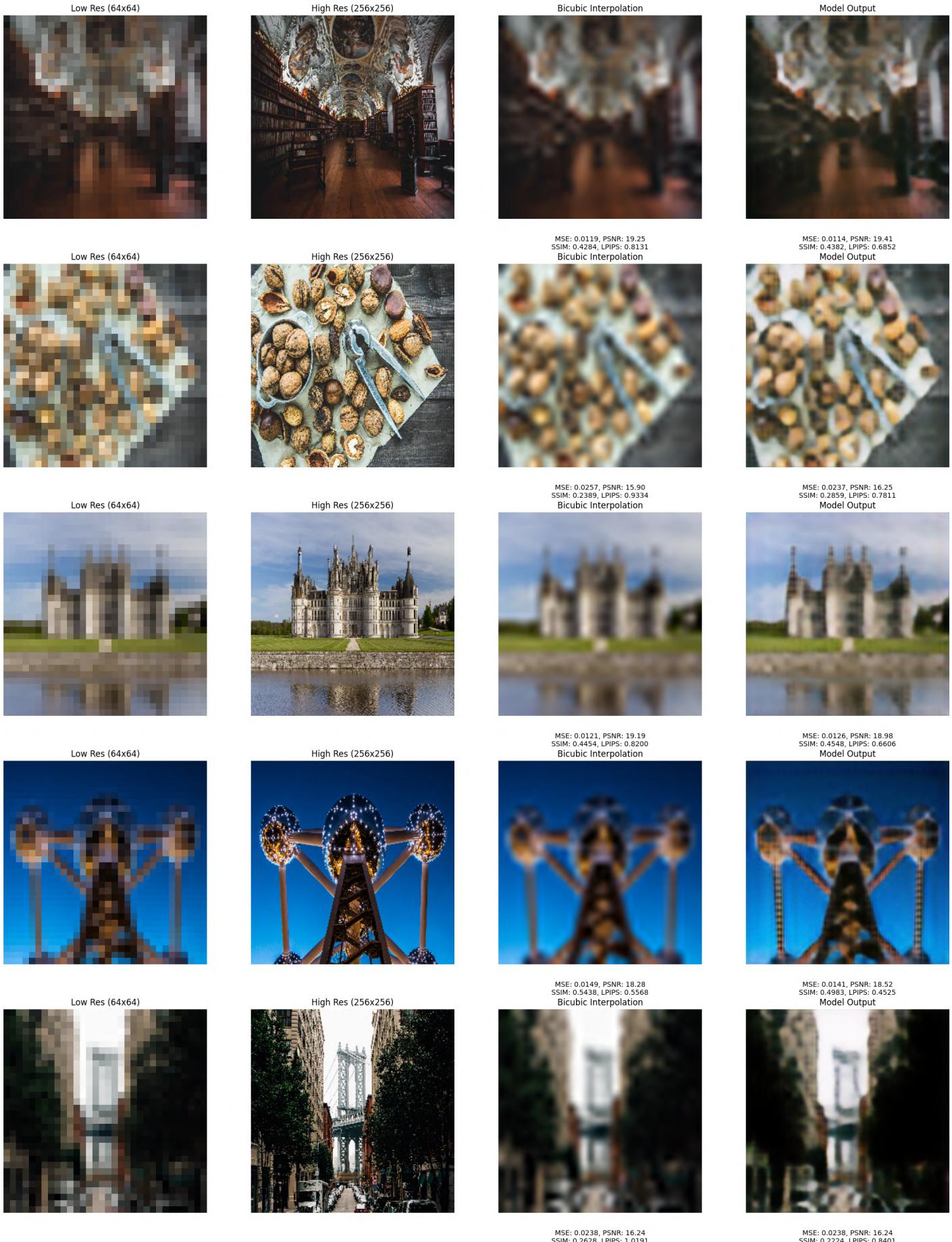
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0167, PSNR: 18.48, SSIM: 0.3862, LPIPS: 0.7725

Model Interpolation:

MSE: 0.0161, PSNR: 18.57, SSIM: 0.4014, LPIPS: 0.6220



```

Epoch 46/50: 100%|██████████| 100/100 [02:07<00:00,  1.28s/it, Loss_D=1.0000, Loss_G=0.0186, D(x)=1.0000, D(G(z))=1.0000]
Epoch 47/50: 100%|██████████| 100/100 [02:08<00:00,  1.29s/it, Loss_D=1.0000, Loss_G=0.0186, D(x)=1.0000, D(G(z))=1.0000]
Epoch 48/50: 100%|██████████| 100/100 [02:08<00:00,  1.29s/it, Loss_D=1.0000, Loss_G=0.0188, D(x)=1.0000, D(G(z))=1.0000]
Epoch 49/50: 100%|██████████| 100/100 [02:08<00:00,  1.28s/it, Loss_D=1.0000, Loss_G=0.0186, D(x)=1.0000, D(G(z))=1.0000]
Epoch 50/50: 100%|██████████| 100/100 [02:07<00:00,  1.28s/it, Loss_D=1.0000, Loss_G=0.0186, D(x)=1.0000, D(G(z))=1.0000]

```

Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]

Loading model from: /home/irek/Development/SIGK/.venv/lib/python3.12/site-packages/lpips/weights/v0.1/alex.pth

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [0.008250146..1.0380698].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.04190735..0.91447693].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.08628853..1.078829].

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-0.0150435455..0.79502785].

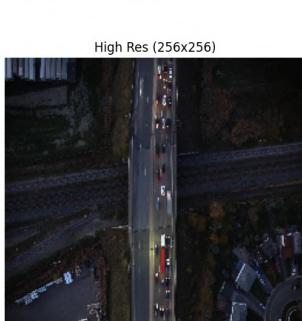
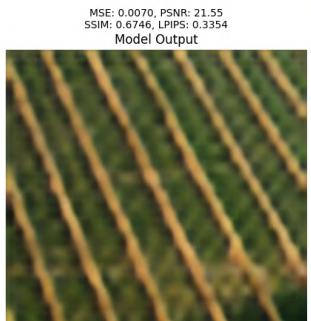
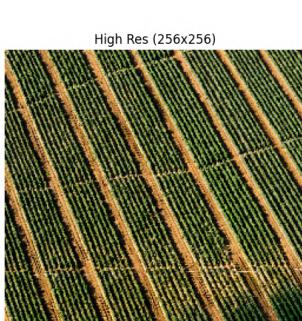
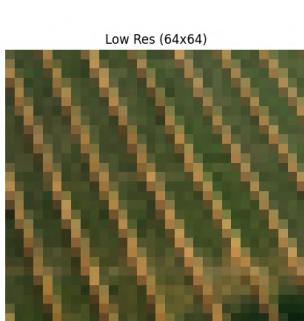
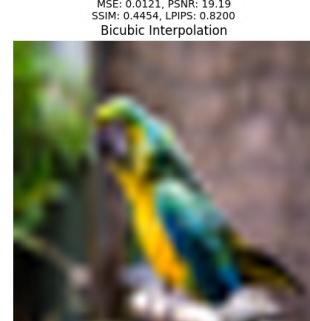
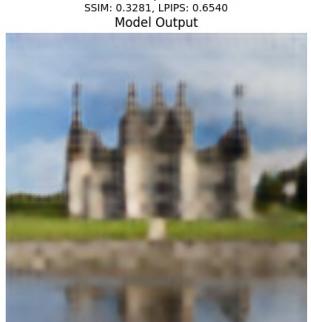
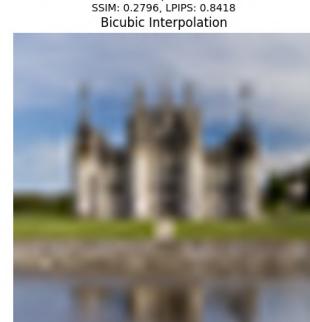
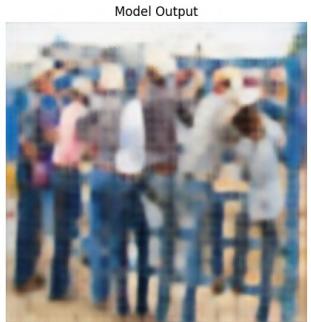
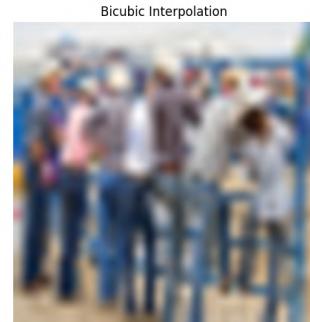
Średnie metryki dla całego datasetu walidacyjnego:

Bicubic Interpolation:

MSE: 0.0167, PSNR: 18.48, SSIM: 0.3862, LPIPS: 0.7725

Model Interpolation:

MSE: 0.0159, PSNR: 18.67, SSIM: 0.4095, LPIPS: 0.6054



MSE: 0.0042, PSNR: 23.76
SSIM: 0.5481, LPIPS: 0.7061

MSE: 0.0041, PSNR: 23.88
SSIM: 0.5474, LPIPS: 0.5245

Podsumowanie

Zaproponowana architektura inspirowana podejściem SRGAN osiąga nieco lepsze wyniki niż interpolacja bikubiczna. Obrazy generowane przez sieć są bardziej ostre niż obrazy interpolowane. Posiadają jednak widoczne artefakty: obramówka, szachownica, zduplikowane krawędzie. Trening przy GAN szybko przestał uwzględniać dyskryminator, który stawał się zbyt słaby i zawsze klasyfikował obrazy jako realistyczne, przez co generator nie otrzymuje gradientów od dyskryminatora i w zasadzie trening staje się zwykłym treningiem generatora a nie GAN. Uczenie dłuższe niż 50 epok mogłoby nieco poprawić generacje, ponieważ metryki cały czas się nieznacznie poprawiały ale zdecydowano zamiast tego podjąć próbę poprawy treningu GAN