FSRCNN 경량화 진행 상황

염지현

1. 가장 가벼운 모델 선택

(first_part): Sequential((0): Conv2d(1, 3, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1)) (1): PReLU(num_parameters=3))
(mid_part): Sequential((0): Conv2d(3, 3, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1)) (1): PReLU(num_parameters=3))
(last_part): ConvTranspose2d(3, 1, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), output_padding=(1, 1)))

CPU									GPU								
		in_ch	out_ch	kernel_size	stride	padding	output_padding	최종 소요 시간			in_ch	out_ch	kernel_size	stride	padding	output_padding	최종 소요 시간
FIRST	conv1	1	3	3	1	1		0.1165	FIRST	conv1	1	3	3	1	1		0.0031
MID	conv1	3	3	3	1	1			MID	conv1	3	3	3	1	1		
	conv2	-	-	-	-	-				conv2	-	-	-	-	-		
	conv3	-	-	-	-	-				conv3	-	-	-	-	-		
	conv4	-	-	-	-	-				conv4	-	-	-	-	-		
	conv5	-	-	-	-	-				conv5	-	-	-	-	-		
	conv6	-	-	-	-	-				conv6	-	-	-	-	-		
LAST	convtransposed	3	1	3	2	1	1		LAST	convtransposed	3	1	3	2	1	1	

Channel, kernel size 대폭 축소

2. augmentation

1. FSRCNN 소요시간 비교

FIRST PART					
C	소요시간	횟수	최종 소요 시간	Python	소요시간
padding	0.006	56	0.336	-	-
getbias	0.001	1	0.001	getbias	0.01
getPReLU	0.001	1	0.001	getPReLU	
getkernel	0.001	56	0.056	getkernel	
convolution	0.158	56	8.848	convolution	0.392
PReLU	0.007	1	0.007	PReLU	0.131
누적			9.249		0.533

MID PART – CONV1 LAYER					
C	소요시간	횟수	최종 소요 시간	Python	소요시간
				Mid part 모델 선언	0.004
				Mid part get total weight	0.003
getbias	0.001	1	0.001		
getPReLU	0.001	1	0.001		
getkernel	0.005	672	3.36		
convoltuion	0.01	672	6.72	convoltuion	0.184
PReLU	0.008	12	0.096	PReLU	0.026
누적			10.178		0.21

1. FSRCNN 소요시간 비교

MID PART – CONV2 LAYER					
C	소요시간	횟수	최종 소요 시간	Python	소요시간
getbias	0.001	1	0.001		
getPReLU	0.001	1	0.001		
getkernel	0.005	144	0.72		
padding	0.005	144	0.72		
convolution	0.065	144	9.36	convolution	0.235
PReLU	0.008	12	0.096	PReLU	0.027
누적			10.898		0.262

MID PART – CONV3 LAYER					
C	소요시간	횟수	최종 소요 시간	Python	소요시간
getbias	0.001	1	0.001		
getPReLU	0.001	1	0.001		
getkernel	0.005	144	0.72		
padding	0.006	144	0.864		
convoltuion	0.066	144	9.504	convoltuion	0.235
PReLU	0.008	12	0.096	PReLU	0.03
누적			11.186		0.265

1. FSRCNN 소요시간 비교

MID PART – CONV4 LAYER					
C	소요시간	횟수	최종 소요 시간	Python	소요시간
getbias	0.001	1	0.001		
getPReLU	0.001	1	0.001		
getkernel	0.005	1	0.005		
padding	0.006	144	0.864		
convoltuion	0.066	144	9.504	convoltuion	0.227
PReLU	0.008	12	0.096	PReLU	0.026
누적			10.471		0.253

MID PART – CONV5 LAYER					
C	소요시간	횟수	최종 소요 시간	Python	소요시간
getbias	0.001	1	0.001		
getPReLU	0.001	1	0.001		
getkernel	0.005	1	0.005		
padding	0.006	144	0.864		
convoltuion	0.066	144	9.504	convoltuion	0.231
PReLU	0.008	12	0.096	PReLU	0.028
누적			10.471		0.259

1. FSRCNN 소요시간 비교

MID PART – CONV6 LAYER					
C	소요시간	횟수	최종 소요 시간	Python	소요시간
getbias	0.001	1	0.001		
getPReLU	0.001	1	0.001		
getkernel	0.004	672	2.688		
convoltuion	0.01	672	6.72	convoltuion	0.18
PReLU	0.008	56	0.448	PReLU	0.198
			9.858		0.378

LAST PART					
C	소요시간	횟수	최종 소요 시간	Python	소요시간
				모델 선언	0.002
getbias	0.001	1	0.001	getbias	0
getkernel	0.005	56	0.28	getkernel	
transpose padding	0.005	56	0.28		
convolution	1.789	56	100.184	convolution	1.466
plus bias	0.02	1	0.02		
			100.765		1.468

2. FSRCNN 속도 개선 아이디어(1)

```
for (int z = 0; z < H; z++) { // z: Height, y축
    for (int t = 0; t < W; t++) { // t: Width, x축
        _element sr[9];
        for (int i = 0; i < ks2_5; i++) { // y축 kernel size
            for (int j = 0; j < ks2_5; j++) { // x축 kernel size
                sr[i * ks2_5 + j].cols = (t + ((W + pad2_5 + pad2_5) * i) + j + (z * (W + pad2_5 + pad2_5)));
                sr[i * ks2_5 + j].w = kernel2_5[i * ks2_5 + j];
            }
        }

        float sumb = 0;
        sumb = 0;
        for (int j = 0; j < ks2_5; j++) {
            for (int k = 0; k < ks2_5; k++) {
                sumb = sumb + sr[j * ks2_5 + k].w * float(Padding2_5[sr[j * ks2_5 + k].cols]);
            }
        }
        TempY[z * W + t] = sumb;
    }
}
```

```
*/
int y2_5 = 0;
int k = W * ks2_5;
for (int z = 0; z < H; z++) { // z: Height, y축
    int h_start = z * PW2_5 + 1;
    int h_end = h_start + k;
    for (int t = 0; t < W; t++) { // t: Width, x축
        int w_start = t + 1;
        int w_end = w_start + ks2_5;
        float sumb = 0;
        int kx = 0;
        int ky = 0;
        for (int i = h_start; i < h_end; i += PW2_5) {
            for (int j = w_start; j < w_end; j++) {
                sumb = sumb + kernel2_5[ky + kx] * float(Padding2_5[i + j]);
                //printf("kernel index: %d, padding[%d]: %f\n", kx + ky, i + j, Padding3[i + j]);
                kx++;

            }
            kx = 0;
            ky += ks2_5;
        }
        TempY[y2_5 + t] = sumb;
    }
    y2_5 += W;
}
```

```
TOTAL TIME: 233.574005
```

```
TOTAL TIME: 186.266006
```

3. 피드백