Deepdeblur on 6490

1. env

- x86
- Ubuntu 18.04
- python == 3.6.13

```
imageio==2.8.0
kiwisolver==1.3.1
matplotlib==3.2.1
numpy==1.18.1
onnxruntime==1.10.0
Pillow==8.4.0
pip==21.2.2
protobuf==3.19.6
pyparsing==3.1.1
readline==6.2.4.1
scikit-image==0.16.2
scipy==1.5.4
torch==1.6.0
tqdm==4.46.1
```

2. Modify & Export

```
git clone https://github.com/SeungjunNah/DeepDeblur-PyTorch.git
```

Download prereained model GOPRO_L1, unzip, and put them under DeepDeblur-PyTorch/experiment.

```
diff --git a/src/model/__init__.py b/src/model/__init__.py
index 850e484..5661053 100644
--- a/src/model/__init__.py
+++ b/src/model/__init__.py
@@ -61,15 +61,16 @@ class Model(nn.Module):
        model_path = os.path.join(self.save_dir, 'model-
{:d}.pt'.format(epoch))
```

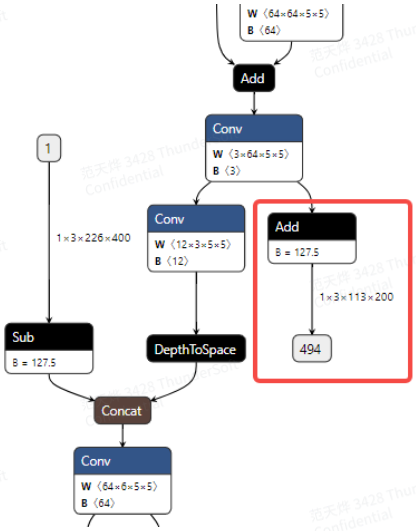
```
        return model_path

-    def state_dict(self):
+    def state_dict(self, destination=None, prefix=' ', keep_vars=True):
+        print("!!!!! cust state_dict called ")
        state_dict = {}
        for model_key in self.model:
            if self.model[model_key] is not None:
                parallelized = isinstance(self.model[model_key], (DataParallel, DistributedDataParallel))
                if parallelized:
-                    state_dict[model_key] = self.model[model_key].module.state_dict()
+                    state_dict[model_key] = self.model[model_key].module.state_dict(destination, prefix, keep_vars=keep_vars)
                else:
-                    state_dict[model_key] = self.model[model_key].state_dict()
+                    state_dict[model_key] = self.model[model_key].state_dict(destination, prefix, keep_vars=keep_vars)

        return state_dict

```

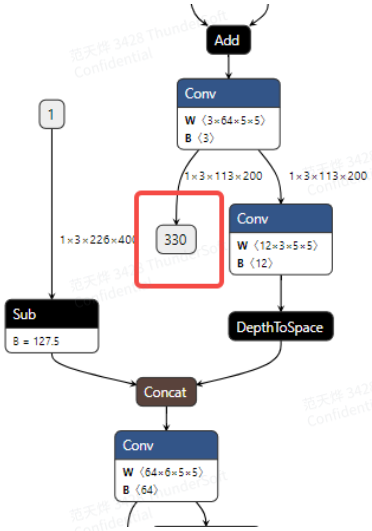
Export model to onnx:

```
diff --git a/src/train.py b/src/train.py
index 43448bc..e15ee74 100644
--- a/src/train.py
+++ b/src/train.py
@@ -145,6 +145,9 @@ class Trainer():
            input, target = data.common.to(
                batch[0], batch[1], device=self.device, dtype=self.dtype_eval)
            with amp.autocast(self.args.amp):
+                with torch.no_grad():
+                    torch.onnx.export(self.model.model.G, input, "deepdeblur.onnx", export_params=True, do_constant_folding=True, opset_version=11)
+                exit()
                output = self.model(input)

            if mode == 'demo':  # remove padded part

```

3. Edit

Use the ONNX editing tool (such as https://github.com/ZhangGe6/onnx-modifier.git), remove the Add operation before the output node. (A total of 3 Add nodes need to be removed)





4. Convert ONNX model into dlc

```
snpe-onnx-to-dlc \
    -i modified_deepdeblur.onnx \
    -o modified_deepdeblur_GOPRO-L1_snpe-2.13.dlc

snpe-dlc-quantize \
    --input_dlc modified_deepdeblur_GOPRO-L1_snpe-2.13.dlc \
    --output_dlc modified_deepdeblur_GOPRO-L1_snpe-2.13_quantize.dlc \
    --input_list input.txt

snpe-dlc-graph-prepare \
    --htp_archs v68 \
    --set_output_tensors "492,411,330" \
    --input_dlc ./modified_deepdeblur_GOPRO-L1_snpe-2.13_quantize.dlc \
    --output_dlc modified_deepdeblur_GOPRO-L1_snpe-2.13_quantize_cached_v68.dlc
    --verbose
```

To quantize model with multiple input nodes, input_list should be like

```
/home/ts/deepdeblur/2.raw /home/ts/deepdeblur/1.raw /home/ts/deepdeblur/0.raw
```