In [160]:

```python
from fastai.vision.all import *
from fastai.data.all import *
from fastai.tabular.all import *
from fastai.vision.all import *
from fastai.metrics import error_rate
from fastai.imports import *
```

In [161]:

```python
print(torch.cuda.device(0))
print(torch.cuda.get_device_name(0))
```

```
<torch.cuda.device object at 0x00000126E81D46D0>
NVIDIA GeForce GTX 1650
```

In [162]:

```python
#################################DATASET###################
```

In [163]:

```python
origin = Path("C:/Development/meteor_detector/dataset/v7_adjusted/dataset/")
origin.ls()
```

Out[163]:

```
(#4) [Path('C:/Development/meteor_detector/dataset/v7_adjusted/dataset/test_set'),Pa
th('C:/Development/meteor_detector/dataset/v7_adjusted/dataset/test_set_NEW'),Path
('C:/Development/meteor_detector/dataset/v7_adjusted/dataset/test_set_OLD'),Path
('C:/Development/meteor_detector/dataset/v7_adjusted/dataset/training_set')]
```

In [164]:

```python
rows_values = []
for dataset in ["test_set", "training_set"]:
    for meteor in ["meteor", "no-meteor"]:
        for fn in (origin/f"{dataset}/{meteor}").glob("*/"):
            rows_values += [(str(fn)[len(str(origin)):], dataset, meteor)]


df = pd.DataFrame(rows_values, columns=["fn", "dataset", "meteor"])
df
```

Out[164]:

	fn	dataset	meteor
0	\test_set\meteor\image-20210219194325.jpg	test_set	meteor
1	\test_set\meteor\image-20210219195726.jpg	test_set	meteor
2	\test_set\meteor\image-20210220000355.jpg	test_set	meteor
3	\test_set\meteor\image-20210220061647.jpg	test_set	meteor
4	\test_set\meteor\image-20210220063819.jpg	test_set	meteor
...	...	...	...
57165	\training_set\no-meteor\image-20210413064838.jpg	training_set	no-meteor
57166	\training_set\no-meteor\image-20210413064908.jpg	training_set	no-meteor
57167	\training_set\no-meteor\image-20210413064938.jpg	training_set	no-meteor
57168	\training_set\no-meteor\image-20210413065008.jpg	training_set	no-meteor
57169	\training_set\no-meteor\image-20210413065038.jpg	training_set	no-meteor

57170 rows × 3 columns

In [165]:
```
df.groupby(["dataset","meteor"]).size()
```

Out[165]:
```
dataset       meteor
test_set      meteor          300
              no-meteor      5008
training_set  meteor         1567
              no-meteor     50295
dtype: int64
```

In [166]:
```
df[df.dataset=="training_set"]
```

Out[166]:

	fn	dataset	meteor
5308	\training_set\meteor\image-20210219194125.jpg	training_set	meteor
5309	\training_set\meteor\image-20210219194155.jpg	training_set	meteor
5310	\training_set\meteor\image-20210219194225.jpg	training_set	meteor
5311	\training_set\meteor\image-20210219194255.jpg	training_set	meteor
5312	\training_set\meteor\image-20210219195756.jpg	training_set	meteor
...	...	...	...
57165	\training_set\no-meteor\image-20210413064838.jpg	training_set	no-meteor
57166	\training_set\no-meteor\image-20210413064908.jpg	training_set	no-meteor
57167	\training_set\no-meteor\image-20210413064938.jpg	training_set	no-meteor
57168	\training_set\no-meteor\image-20210413065008.jpg	training_set	no-meteor
57169	\training_set\no-meteor\image-20210413065038.jpg	training_set	no-meteor

51862 rows × 3 columns

In [167]:
```
#Balanceo de los grupos de train y test
df_no_meteor_train = df[(df["meteor"]=="no-meteor") & (df["dataset"]=="training_set")].sample(n=1567) #frac=0.1
df_no_meteor_test  = df[(df["meteor"]=="no-meteor") & (df["dataset"]=="test_set")].sample(n=1000)
df_meteor_train    = df[(df["meteor"]=="meteor")    & (df["dataset"]=="training_set")]
df_meteor_test     = df[(df["meteor"]=="meteor")    & (df["dataset"]=="test_set")]

df=pd.concat([df_no_meteor_train,df_no_meteor_test,df_meteor_train,df_meteor_test])
df.groupby(["dataset","meteor"]).size()
```

Out[167]:
```
dataset       meteor
test_set      meteor        300
              no-meteor    1000
training_set  meteor       1567
              no-meteor    1567
dtype: int64
```

In [168]:
```
df.groupby(["dataset","meteor"]).size()
```

Out[168]:
```
dataset       meteor
test_set      meteor        300
              no-meteor    1000
training_set  meteor       1567
              no-meteor    1567
dtype: int64
```

In [169]:

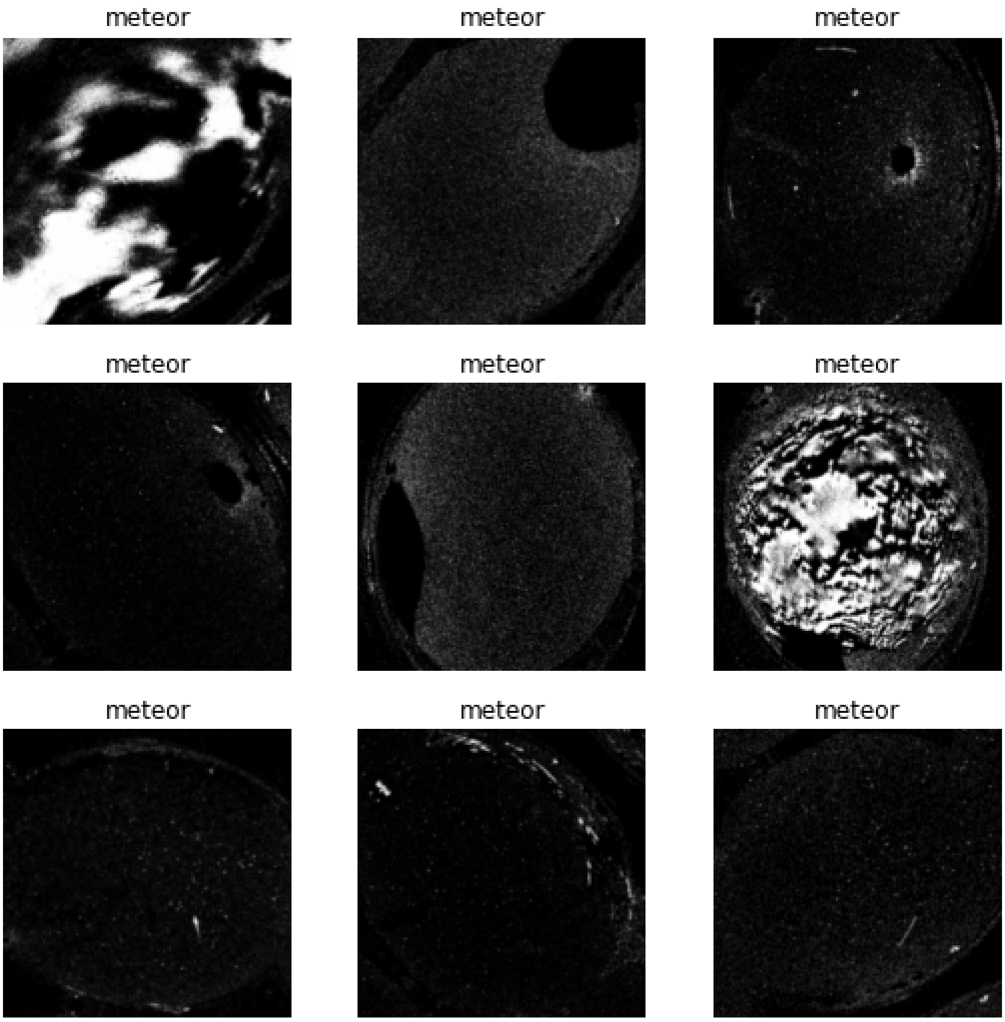
```
dls = ImageDataLoaders.from_df(df[df["dataset"]!="test_set"],
                               folder=origin ,
                               bs=32,
                               batch_tfms=aug_transforms(max_rotate=180,max_warp=0,max_zoom=0),

                               item_tfms=[Resize(224)],
                               fn_col=0,
                               label_col=2,
                               shuffle_train=True,
                               drop_last=True,
                               valid_pct=0.2,
                               num_workers=0)
```

In [170]:

```
#aug_transforms?
```

In [172]:

```
dls.show_batch()
```



In [173]:

```
learn = load_learner("C:/Development/meteor_detector/dataset/v7_adjusted/model_test_nou_resnet34.pkl")
```

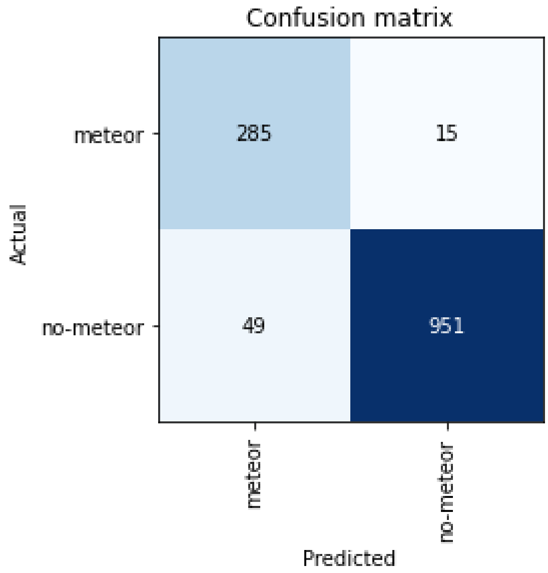
In [174]:

```
interp = ClassificationInterpretation.from_learner(learn, dl=learn.dls.test_dl(df[df[
"dataset"]=="test_set"], with_labels=True, bs=100))
interp.print_classification_report()
```

```
              precision    recall  f1-score   support

      meteor       0.85      0.95      0.90       300
   no-meteor       0.98      0.95      0.97      1000

    accuracy                           0.95      1300
   macro avg       0.92      0.95      0.93      1300
weighted avg       0.95      0.95      0.95      1300
```

In [175]:

```
interp.plot_confusion_matrix(figsize=(4, 4))
```

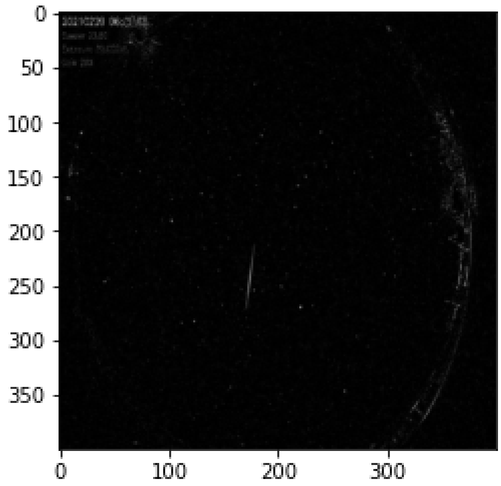


In [184]:

```
img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/eliminat_fond
o/positive_1.jpg')
plt.imshow(img)
```

Out[184]:

```
<matplotlib.image.AxesImage at 0x126ada814f0>
```



In [185]:

```
learn.predict(img)
```

Out[185]:

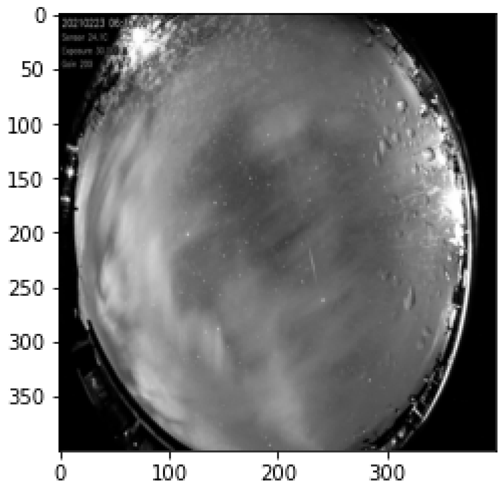
```
('meteor', tensor(0), tensor([9.9942e-01, 5.8004e-04]))
```

In [186]:

```
img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/tractades/positive_4.jpg')
plt.imshow(img)
```

Out[186]:

```
<matplotlib.image.AxesImage at 0x126bc1d31f0>
```



In [187]:

```
learn.predict(img)
```

Out[187]:

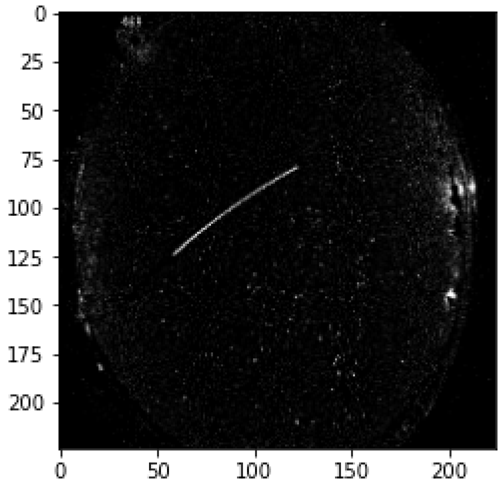
```
('meteor', tensor(0), tensor([0.9925, 0.0075]))
```

In [188]:

```
img = PILImage.create('C:/Development/meteor_detector/dataset/vDef_dia0504/test/my2.jpg')
plt.imshow(img)
```

Out[188]:

```
<matplotlib.image.AxesImage at 0x126ada731f0>
```



In [189]:

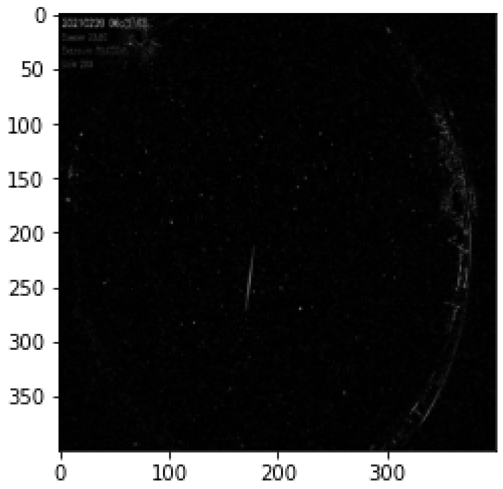
```
learn.predict(img)
```

Out[189]:

```
('meteor', tensor(0), tensor([9.9960e-01, 3.9926e-04]))
```

```
In [190]: img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/eliminat_fondo/positive_1.jpg')
          plt.imshow(img)
```

```
Out[190]: <matplotlib.image.AxesImage at 0x12682aab310>
```

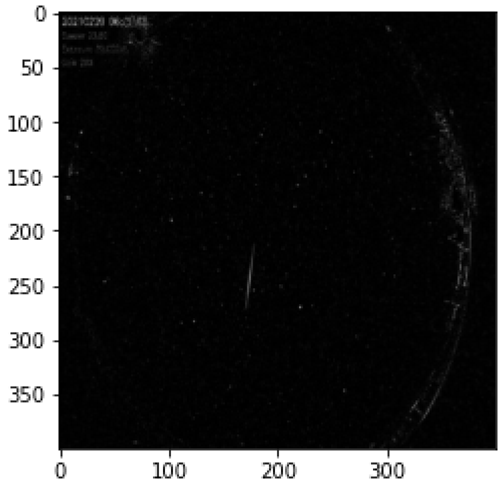


```
In [191]: learn.predict(img)
```

```
Out[191]: ('meteor', tensor(0), tensor([9.9942e-01, 5.8004e-04]))
```

```
In [192]: img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/eliminat_fondo/positive_1.jpg')
          plt.imshow(img)
```

```
Out[192]: <matplotlib.image.AxesImage at 0x126a0c6f6d0>
```



```
In [193]: learn.predict(img)
```

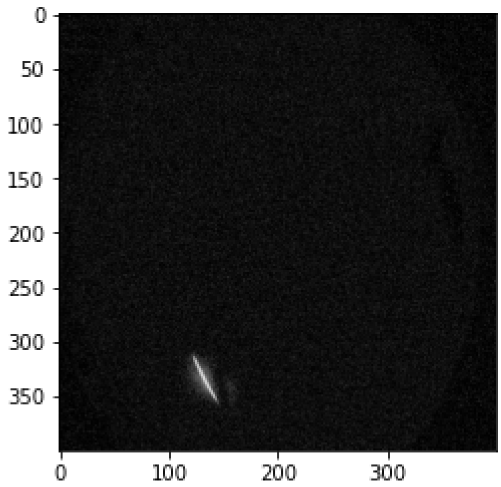
```
Out[193]: ('meteor', tensor(0), tensor([9.9942e-01, 5.8004e-04]))
```

In [194]:

```
img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/eliminat_fondo/positive_3.jpg')
plt.imshow(img)
```

Out[194]:

```
<matplotlib.image.AxesImage at 0x126a205f880>
```



In [195]:

```
learn.predict(img)
```

Out[195]:

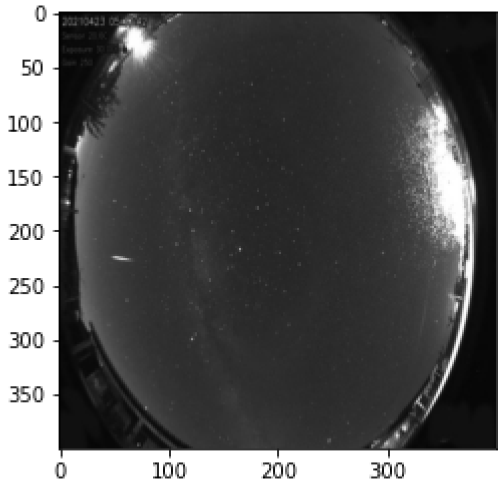
```
('meteor', tensor(0), tensor([0.9870, 0.0130]))
```

In [196]:

```
img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/eliminat_fondo/positive_14.jpg')
plt.imshow(img)
```

Out[196]:

```
<matplotlib.image.AxesImage at 0x126a1fb5370>
```



In [197]:

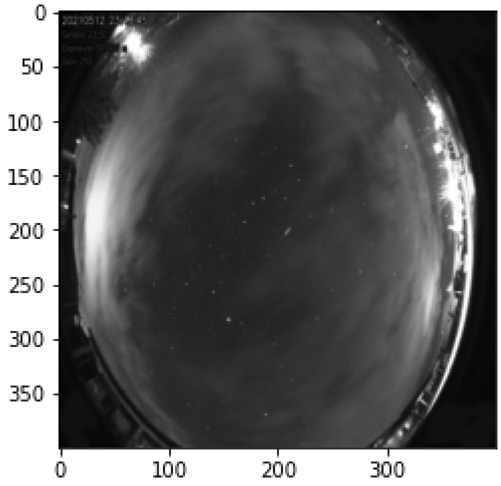
```
learn.predict(img)
```

Out[197]:

```
('meteor', tensor(0), tensor([9.9983e-01, 1.6537e-04]))
```

```
In [198]: img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/eliminat_fondo/positive_18.jpg')
          plt.imshow(img)
```

```
Out[198]: <matplotlib.image.AxesImage at 0x12682b2ea30>
```

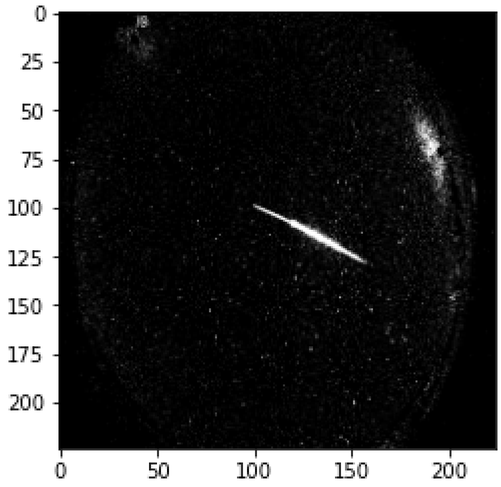


```
In [199]: learn.predict(img)
```

```
Out[199]: ('meteor', tensor(0), tensor([0.9983, 0.0017]))
```

```
In [200]: img = PILImage.create('C:/Development/meteor_detector/dataset/vDef_dia0504/test/my1.jpg')
          plt.imshow(img)
```

```
Out[200]: <matplotlib.image.AxesImage at 0x126a2539130>
```



```
In [201]: learn.predict(img)
```

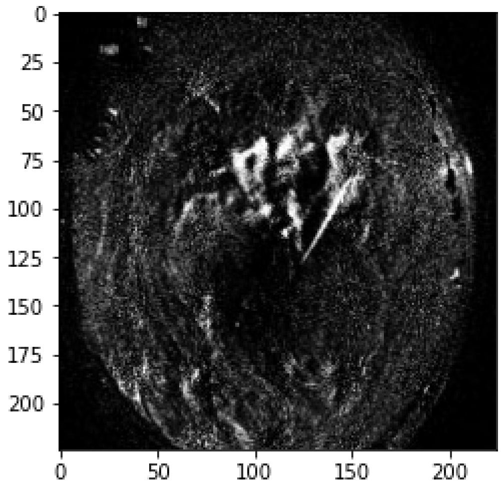
```
Out[201]: ('meteor', tensor(0), tensor([9.9987e-01, 1.3328e-04]))
```

In [202]:

```
#Detection during pilot
img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/pilot_detection.jpg')
plt.imshow(img)
```

Out[202]:

```
<matplotlib.image.AxesImage at 0x126a22e2bb0>
```



In [203]:

```
learn.predict(img)
```

Out[203]:

```
('meteor', tensor(0), tensor([0.9039, 0.0961]))
```

In [204]:

```
print(learn.model)
```

```
Sequential(
  (0): Sequential(
    (0): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=Fals
e)
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=T
rue)
    (2): ReLU(inplace=True)
    (3): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
    (4): Sequential(
      (0): BasicBlock(
        (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
      (1): BasicBlock(
        (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
      (2): BasicBlock(
        (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
      )
    )
    (5): Sequential(
      (0): BasicBlock(
        (conv1): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        (downsample): Sequential(
          (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        )
      )
      (1): BasicBlock(
        (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
```

```
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
      (2): BasicBlock(
        (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
      (3): BasicBlock(
        (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
    )
    (6): Sequential(
      (0): BasicBlock(
        (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        (downsample): Sequential(
          (0): Conv2d(128, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        )
      )
      (1): BasicBlock(
        (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
      (2): BasicBlock(
        (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
      (3): BasicBlock(
        (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
      (4): BasicBlock(
        (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
      (5): BasicBlock(
        (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
    )
    (7): Sequential(
      (0): BasicBlock(
        (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1),
bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        (downsample): Sequential(
          (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
          (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        )
      )
      (1): BasicBlock(
        (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
```

```
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
      (2): BasicBlock(
        (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
        (relu): ReLU(inplace=True)
        (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
        (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_
stats=True)
      )
    )
  )
  (1): Sequential(
    (0): AdaptiveConcatPool2d(
      (ap): AdaptiveAvgPool2d(output_size=1)
      (mp): AdaptiveMaxPool2d(output_size=1)
    )
    (1): Flatten(full=False)
    (2): BatchNorm1d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats
=True)
    (3): Dropout(p=0.25, inplace=False)
    (4): Linear(in_features=1024, out_features=512, bias=False)
    (5): ReLU(inplace=True)
    (6): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=
True)
    (7): Dropout(p=0.5, inplace=False)
    (8): Linear(in_features=512, out_features=2, bias=False)
  )
)
```

In [ ]: