In [1]:
```
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 [2]:
```
print(torch.cuda.device(0))
print(torch.cuda.get_device_name(0))
```

```
<torch.cuda.device object at 0x000001E471D84640>
NVIDIA GeForce GTX 1650
```

In [3]:
```
##################################DATASET###################
```

In [4]:
```
origin = Path("C:/Development/meteor_detector/dataset/v7_adjusted/dataset/")
```

In [5]:
```
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[5]:

	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 [6]:
```
df.groupby(["dataset","meteor"]).size()
```

Out[6]:
```
dataset       meteor
test_set      meteor         300
              no-meteor     5008
training_set  meteor        1567
              no-meteor    50295
dtype: int64
```

In [7]:
```
df.to_csv("C:/Development/meteor_detector/dataset/index.csv", index=False)
```

In [8]:
```
df[df.dataset=="training_set"]
```

Out[8]:

	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 [9]:
```
#Train Dataset Balancing
df_no_meteor_train = df[(df["meteor"]=="no-meteor") & (df["dataset"]=="training_set"
)].sample(n=1567)
df_no_meteor_test  = df[(df["meteor"]=="no-meteor") & (df["dataset"]=="test_set")].sa
mple(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])
```

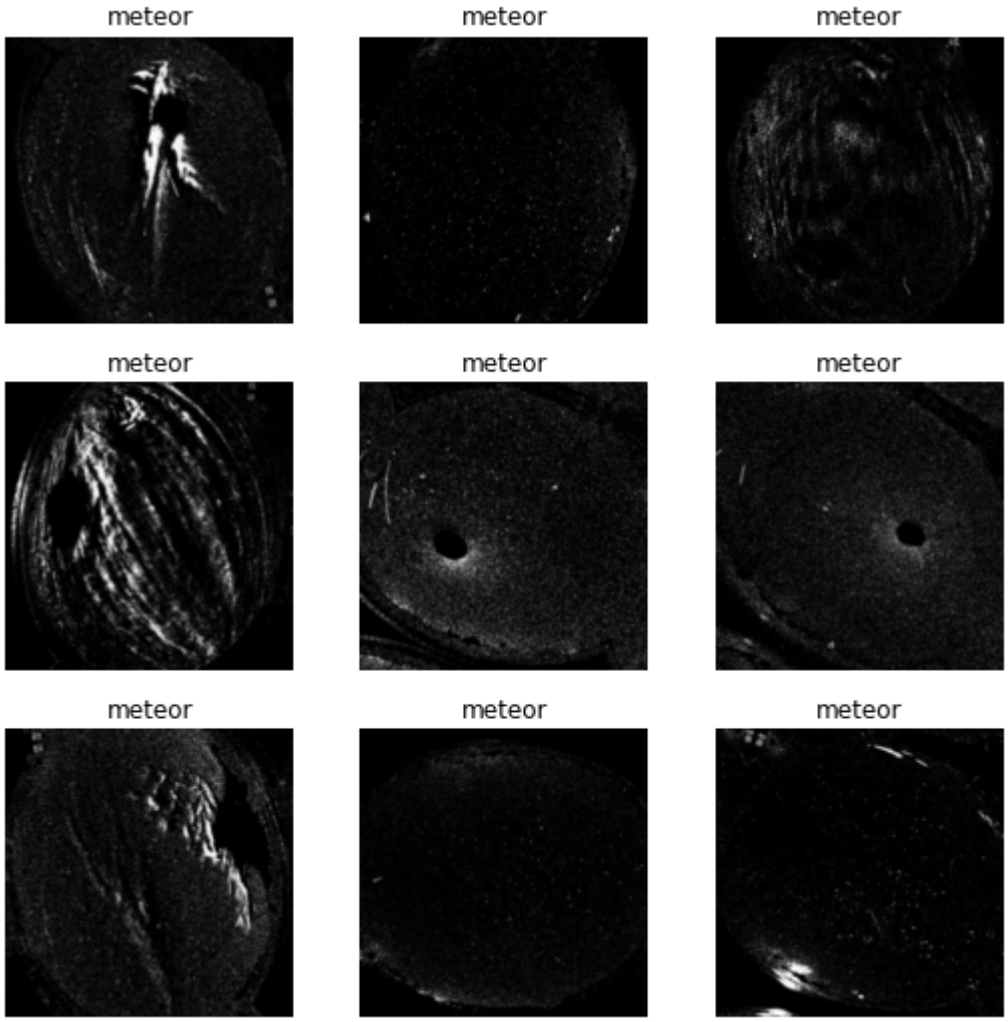
In [10]:
```
df.groupby(["dataset","meteor"]).size()
```

Out[10]:
```
dataset       meteor
test_set      meteor        300
              no-meteor    1000
training_set  meteor       1567
              no-meteor    1567
dtype: int64
```

In [40]:
```
dls = ImageDataLoaders.from_df(df[df["dataset"]!="test_set"],
                               folder=origin ,
                               bs=32,
                               batch_tfms=aug_transforms(max_rotate=180,max_warp=0,ma
x_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 [32]:
```
ImageDataLoaders.from_df??
```

In [42]:
```
dls.show_batch()
```



In [43]:
```
learn = cnn_learner(dls, resnet18,
                    metrics=[error_rate, accuracy, Precision(average='macro'), F1Score(av
erage='macro')],
                     cbs=[EarlyStoppingCallback(patience=25), ShowGraphCallback()]
                    )
```

In [44]:
```
print(learn.dls.train.after_batch)
print(learn.dls.valid.after_batch)
```

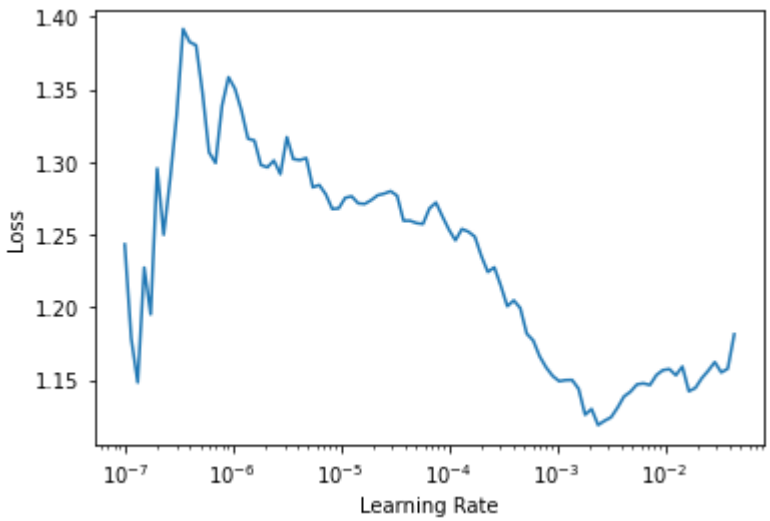
```
Pipeline: IntToFloatTensor -- {'div': 255.0, 'div_mask': 1} -> Flip -- {'size': Non
e, 'mode': 'bilinear', 'pad_mode': 'reflection', 'mode_mask': 'nearest', 'align_corn
ers': True, 'p': 0.5} -> Brightness -- {'max_lighting': 0.2, 'p': 1.0, 'draw': None,
'batch': False} -> Normalize -- {'mean': tensor([[[[0.4850]],

         [[0.4560]],

         [[0.4060]]]], device='cuda:0'), 'std': tensor([[[[0.2290]],

         [[0.2240]],

         [[0.2250]]]], device='cuda:0'), 'axes': (0, 2, 3)}
Pipeline: IntToFloatTensor -- {'div': 255.0, 'div_mask': 1} -> Flip -- {'size': Non
e, 'mode': 'bilinear', 'pad_mode': 'reflection', 'mode_mask': 'nearest', 'align_corn
ers': True, 'p': 0.5} -> Brightness -- {'max_lighting': 0.2, 'p': 1.0, 'draw': None,
'batch': False} -> Normalize -- {'mean': tensor([[[[0.4850]],

         [[0.4560]],

         [[0.4060]]]], device='cuda:0'), 'std': tensor([[[[0.2290]],

         [[0.2240]],

         [[0.2250]]]], device='cuda:0'), 'axes': (0, 2, 3)}
```

In [45]:
```
learn.loss_func
```

Out[45]:
```
FlattenedLoss of CrossEntropyLoss()
```

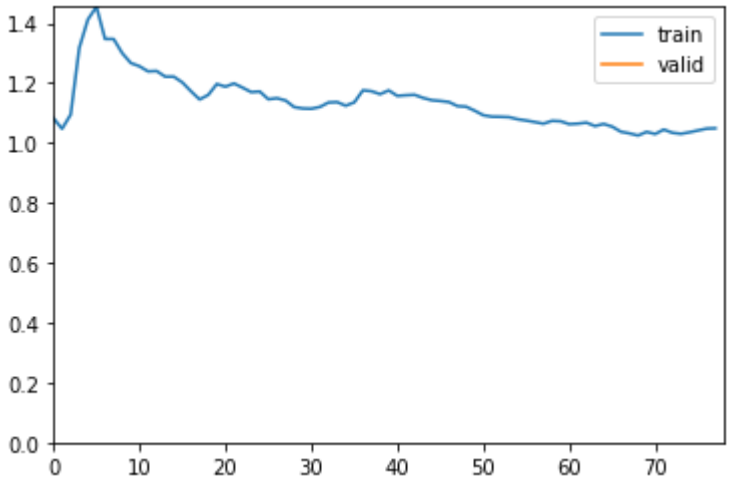
In [50]:
```
learn.lr_find(end_lr=0.1)
```

Out[50]:
```
SuggestedLRs(lr_min=0.00023988329339772463, lr_steep=5.248074330665986e-07)
```

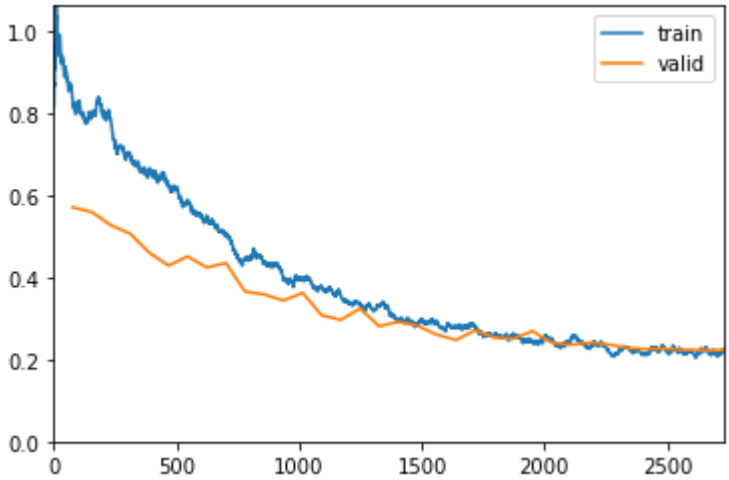


In [51]:
```
learn.fine_tune(35,1*1e-3)
```

epoch	train_loss	valid_loss	error_rate	accuracy	precision_score	f1_score	time
0	1.049252	0.721333	0.359425	0.640575	0.649180	0.639315	00:37



epoch	train_loss	valid_loss	error_rate	accuracy	precision_score	f1_score	time
0	0.838610	0.571901	0.281150	0.718850	0.723393	0.713942	00:41
1	0.803993	0.560540	0.249201	0.750799	0.750238	0.749877	00:41
2	0.782298	0.528214	0.257188	0.742812	0.742992	0.741093	00:41
3	0.698753	0.507604	0.226837	0.773163	0.773163	0.773015	00:41
4	0.654818	0.461071	0.193291	0.806709	0.808311	0.805206	00:41
5	0.628972	0.429607	0.185304	0.814696	0.816974	0.813092	00:41
6	0.584568	0.452043	0.220447	0.779553	0.778983	0.779111	00:43
7	0.552143	0.425085	0.193291	0.806709	0.806176	0.806294	00:41
8	0.504984	0.435844	0.191693	0.808307	0.808307	0.808181	00:41
9	0.447361	0.366071	0.175719	0.824281	0.823795	0.823978	00:41
10	0.437134	0.359059	0.164537	0.835463	0.843530	0.832936	00:44
11	0.398034	0.344151	0.150160	0.849840	0.851551	0.848797	00:44
12	0.400914	0.363281	0.166134	0.833866	0.845921	0.833431	00:41
13	0.378998	0.308167	0.151757	0.848243	0.847817	0.847870	00:40
14	0.356866	0.296856	0.134185	0.865815	0.866439	0.865780	00:41
15	0.335868	0.324765	0.153355	0.846645	0.852948	0.846545	00:43
16	0.329427	0.281342	0.115016	0.884984	0.884582	0.884815	00:41
17	0.293908	0.292421	0.137380	0.862620	0.863808	0.862607	00:41
18	0.284020	0.282753	0.116613	0.883387	0.883129	0.883062	00:43
19	0.294557	0.261082	0.113419	0.886581	0.888582	0.885824	00:45
20	0.283444	0.247845	0.102236	0.897764	0.897943	0.897712	00:44
21	0.280663	0.271215	0.118211	0.881789	0.883700	0.881788	00:42
22	0.258704	0.254404	0.100639	0.899361	0.899153	0.899268	00:45
23	0.262213	0.250810	0.102236	0.897764	0.898987	0.897754	00:43
24	0.245186	0.269664	0.100639	0.899361	0.899153	0.899268	00:41
25	0.249133	0.241461	0.089457	0.910543	0.910431	0.910278	00:40
26	0.247837	0.236015	0.087859	0.912141	0.911933	0.912060	00:40
27	0.226847	0.241507	0.086262	0.913738	0.914132	0.913706	00:41
28	0.221255	0.236792	0.086262	0.913738	0.913355	0.913611	00:40
29	0.227504	0.228003	0.084665	0.915335	0.915619	0.915299	00:41
30	0.215469	0.225727	0.081470	0.918530	0.918324	0.918455	00:42
31	0.219962	0.226292	0.084665	0.915335	0.914950	0.915178	00:45
32	0.218590	0.223887	0.081470	0.918530	0.918168	0.918420	00:48
33	0.226887	0.223008	0.083067	0.916933	0.916842	0.916687	00:41
34	0.220581	0.223404	0.079872	0.920128	0.919792	0.920029	00:41

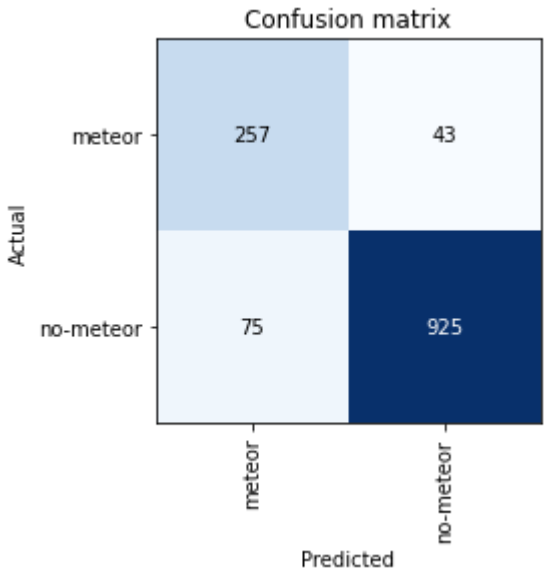


In [52]:
```
#Load or Save Model
#learn.export('C:/Development/guaita/guaita/guAIta_latest_version.pkl')
#learn = load_learner('C:/Development/guaita/guaita/guAIta_latest_version.pkl')
```

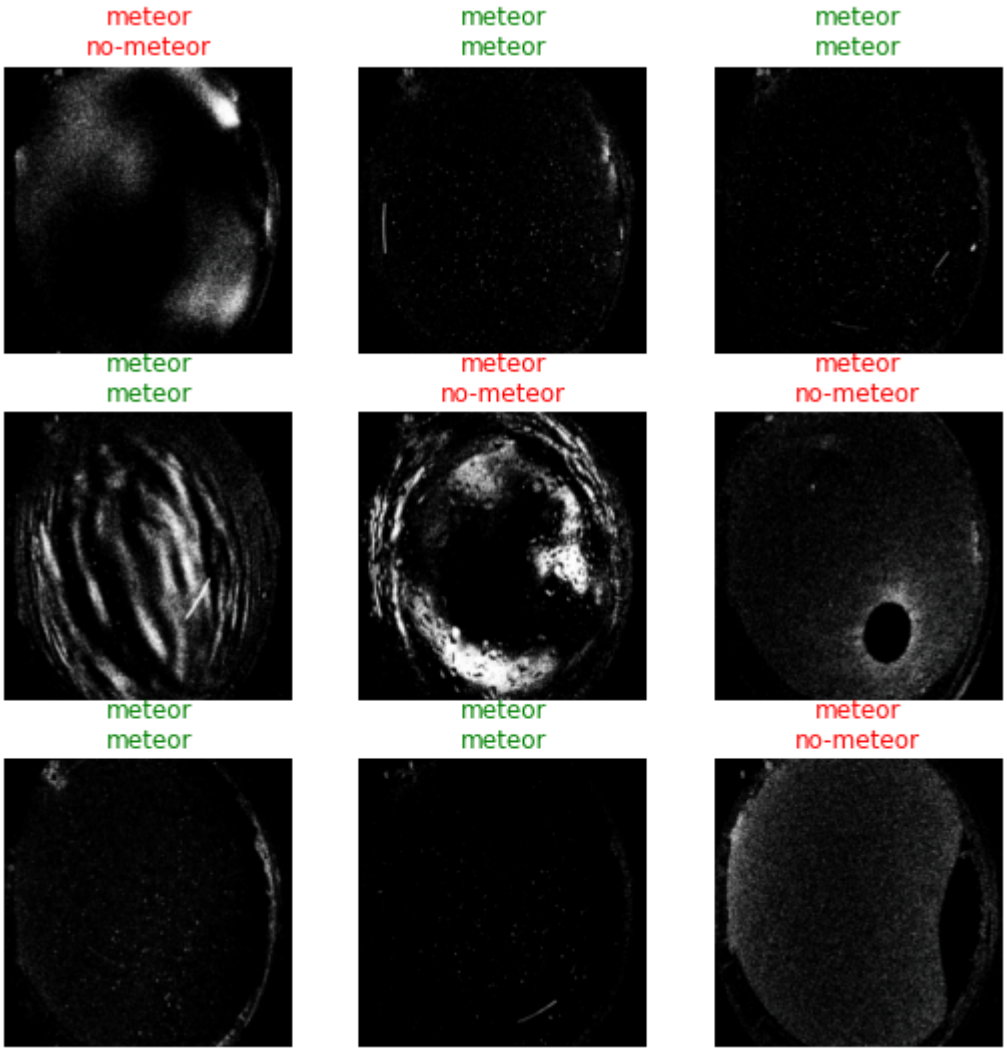
In [53]:
```
interp = ClassificationInterpretation.from_learner(learn, dl=learn.dls.test_dl(df[df[
"dataset"]=="test_set"], with_labels=True, bs=100))
interp.print_classification_report()
```

	precision	recall	f1-score	support
meteor	0.77	0.86	0.81	300
no-meteor	0.96	0.93	0.94	1000
accuracy			0.91	1300
macro avg	0.86	0.89	0.88	1300
weighted avg	0.91	0.91	0.91	1300

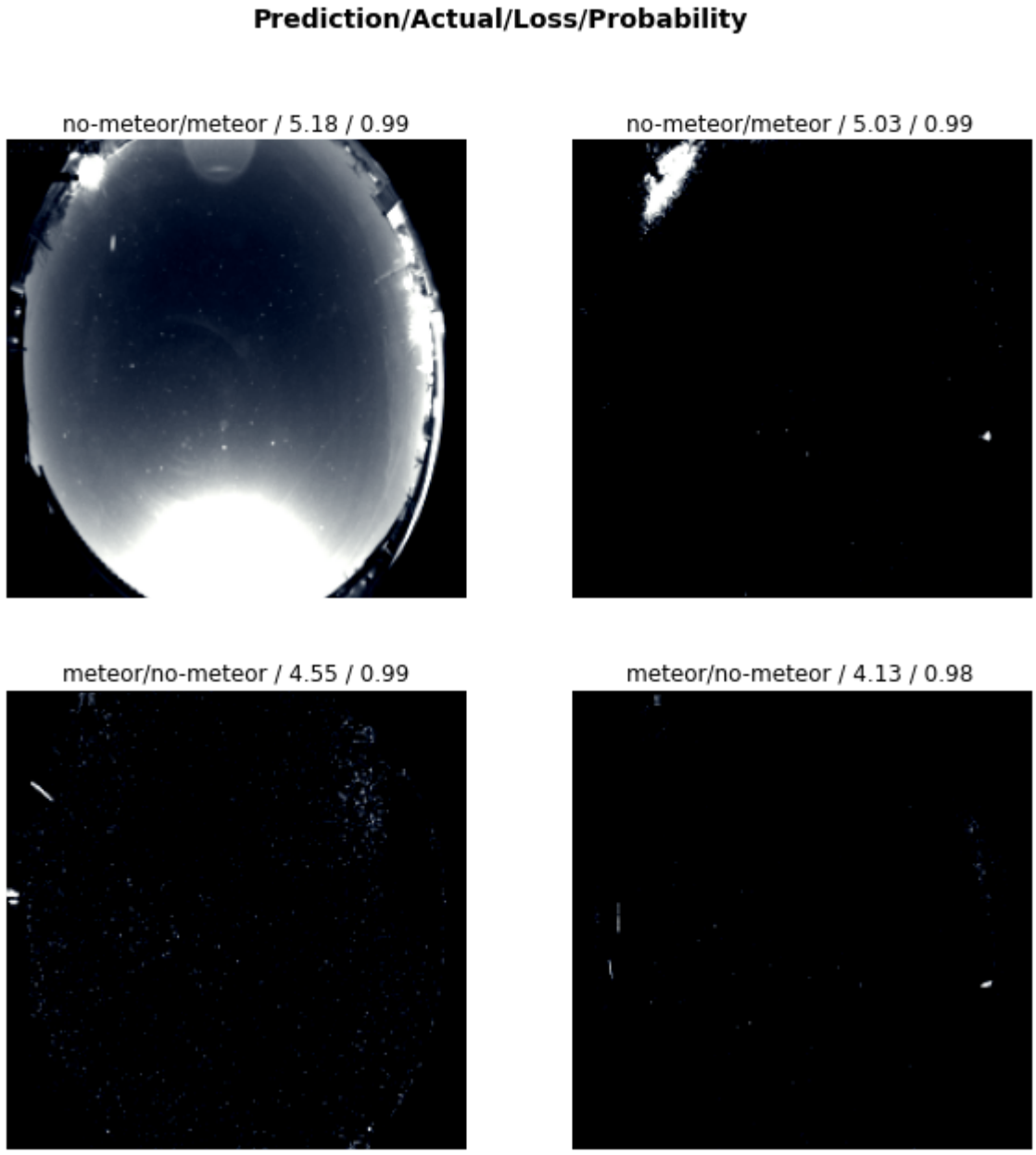
In [54]:
```
interp.plot_confusion_matrix(figsize=(4, 4))
```



In [55]:
```
learn.show_results()
```



In [57]:
```
interp.plot_top_losses(k=4,figsize=(10,10))
```



In [58]:
```
losses,idxs = interp.top_losses(10)
df[df["dataset"]=="test_set"].iloc[idxs]
```

Out[58]:

	fn	dataset	meteor
283	\test_set\meteor\positive_10.jpg	test_set	meteor
15	\test_set\meteor\image-20210315040525.jpg	test_set	meteor
2734	\test_set\no-meteor\image-20210423045608.jpg	test_set	no-meteor
1152	\test_set\no-meteor\image-20210419041253.jpg	test_set	no-meteor
386	\test_set\no-meteor\image-20210418214034.jpg	test_set	no-meteor
4779	\test_set\no-meteor\image-20210505015234.jpg	test_set	no-meteor
1170	\test_set\no-meteor\image-20210419042324.jpg	test_set	no-meteor
1155	\test_set\no-meteor\image-20210419041423.jpg	test_set	no-meteor
284	\test_set\meteor\positive_11.jpg	test_set	meteor
272	\test_set\meteor\image-20210505041221.jpg	test_set	meteor

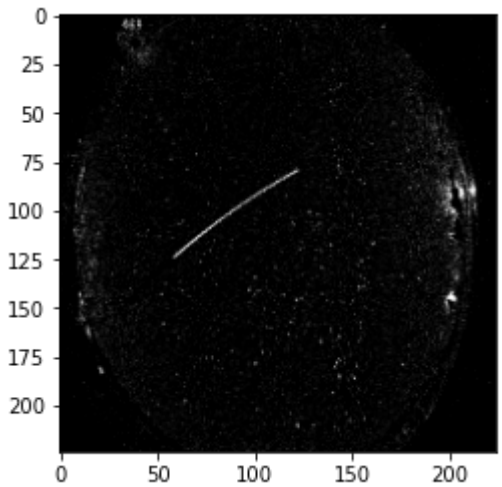
In [38]:
```
#Model Evaluation
```

In [60]:
```
class Hook():
    def __init__(self, m):
        self.hook = m.register_forward_hook(self.hook_func)
    def hook_func(self, m, i, o): self.stored = o.detach().clone()
    def __enter__(self, *args): return self
    def __exit__(self, *args): self.hook.remove()
```

In [61]:
```
class HookBwd():
    def __init__(self, m):
        self.hook = m.register_backward_hook(self.hook_func)
    def hook_func(self, m, gi, go): self.stored = go[0].detach().clone()
    def __enter__(self, *args): return self
    def __exit__(self, *args): self.hook.remove()
```

In [84]:
```
img = PILImage.create('C:/Development/meteor_detector/dataset/vDef_dia0504/test/my2.jpg')
x, = first(dls.test_dl([img]))
plt.imshow(img)
```

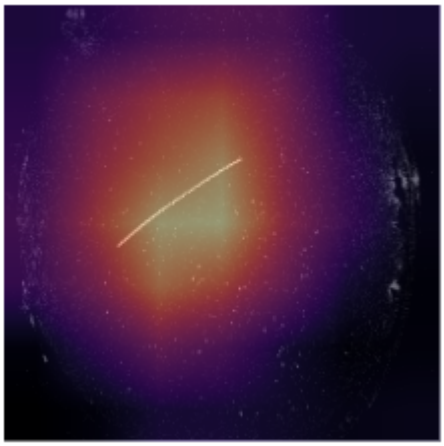
Out[84]:
```
<matplotlib.image.AxesImage at 0x1e448b9dca0>
```



In [85]:
```
cls = 0
with HookBwd(learn.model[0]) as hookg:
    with Hook(learn.model[0]) as hook:
        output = learn.model.eval()(x.cuda())
        act = hook.stored
    output[0,cls].backward()
    grad = hookg.stored
```

In [86]:
```
w = grad[0].mean(dim=[1,2], keepdim=True)
cam_map = (w * act[0]).sum(0)
```

In [87]:
```
x_dec = TensorImage(dls.train.decode((x,))[0][0])
_,ax = plt.subplots()
x_dec.show(ctx=ax)
ax.imshow(cam_map.detach().cpu(), alpha=0.6, extent=(0,224,224,0),
              interpolation='bilinear', cmap='magma');
```

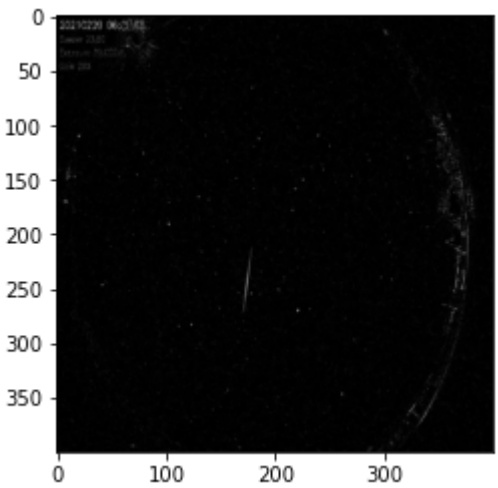


In [88]:
```
learn.predict(img)
```

Out[88]:
```
('meteor', tensor(0), tensor([0.9416, 0.0584]))
```

In [89]:
```
img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/eliminat_fondo/positive_1.jpg')
x, = first(dls.test_dl([img]))
plt.imshow(img)
```

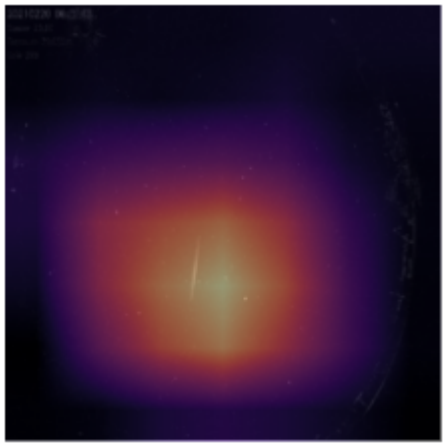
Out[89]:
```
<matplotlib.image.AxesImage at 0x1e444340cd0>
```



In [90]:
```
cls = 0
with HookBwd(learn.model[0]) as hookg:
    with Hook(learn.model[0]) as hook:
        output = learn.model.eval()(x.cuda())
        act = hook.stored
    output[0,cls].backward()
    grad = hookg.stored
```

In [91]:
```
w = grad[0].mean(dim=[1,2], keepdim=True)
cam_map = (w * act[0]).sum(0)
```

In [92]:
```
x_dec = TensorImage(dls.train.decode((x,))[0][0])
_,ax = plt.subplots()
x_dec.show(ctx=ax)
ax.imshow(cam_map.detach().cpu(), alpha=0.6, extent=(0,224,224,0),
              interpolation='bilinear', cmap='magma');
```

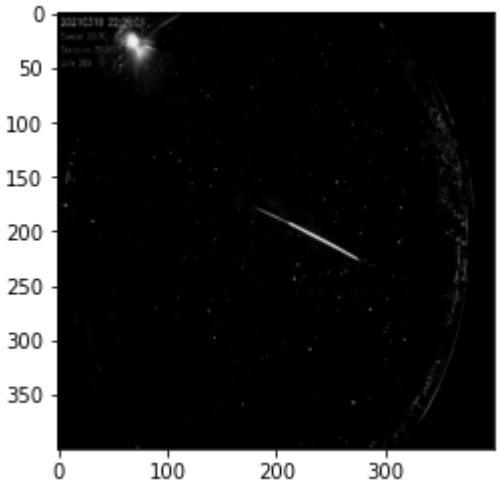


In [93]:
```
learn.predict(img)
```

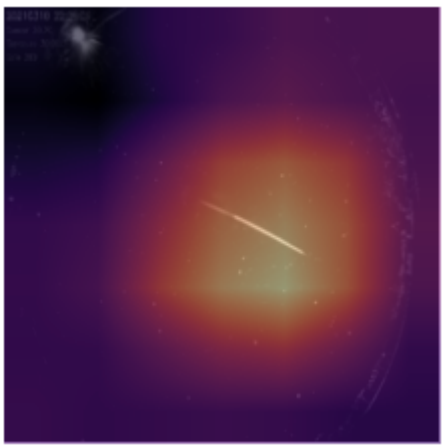
Out[93]:
```
('meteor', tensor(0), tensor([9.9991e-01, 9.2871e-05]))
```

In [94]:
```
img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/eliminat_fondo/positive_2.jpg')
x, = first(dls.test_dl([img]))
plt.imshow(img)
```

Out[94]:
```
<matplotlib.image.AxesImage at 0x1e46ca72790>
```



In [95]:
```
cls = 0
with HookBwd(learn.model[0]) as hookg:
    with Hook(learn.model[0]) as hook:
        output = learn.model.eval()(x.cuda())
        act = hook.stored
    output[0,cls].backward()
    grad = hookg.stored
w = grad[0].mean(dim=[1,2], keepdim=True)
cam_map = (w * act[0]).sum(0)
x_dec = TensorImage(dls.train.decode((x,))[0][0])
_,ax = plt.subplots()
x_dec.show(ctx=ax)
ax.imshow(cam_map.detach().cpu(), alpha=0.6, extent=(0,224,224,0),
              interpolation='bilinear', cmap='magma');
```

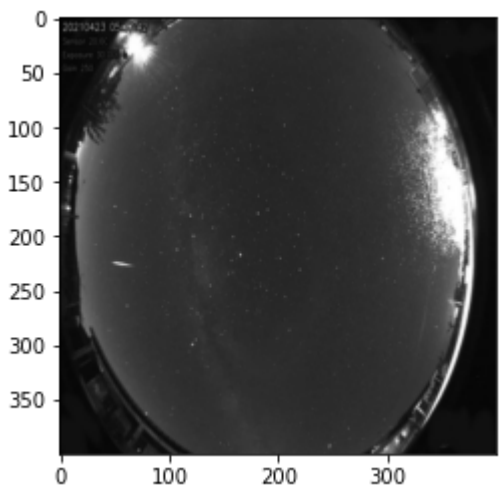


In [96]:
```
learn.predict(img)
```

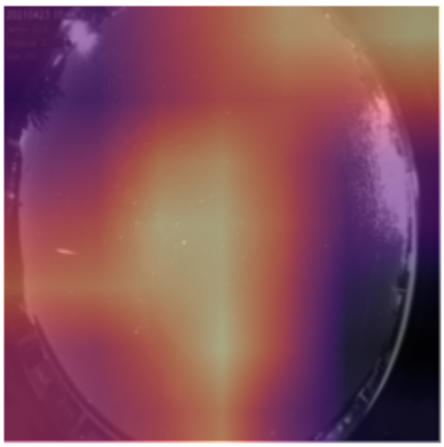
Out[96]:
```
('meteor', tensor(0), tensor([0.9426, 0.0574]))
```

In [97]:
```
img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/eliminat_fondo/positive_14.jpg')
x, = first(dls.test_dl([img]))
plt.imshow(img)
```

Out[97]:
```
<matplotlib.image.AxesImage at 0x1e48ab95640>
```



In [98]:
```
cls = 0
with HookBwd(learn.model[0]) as hookg:
    with Hook(learn.model[0]) as hook:
        output = learn.model.eval()(x.cuda())
        act = hook.stored
    output[0,cls].backward()
    grad = hookg.stored
w = grad[0].mean(dim=[1,2], keepdim=True)
cam_map = (w * act[0]).sum(0)
x_dec = TensorImage(dls.train.decode((x,))[0][0])
_,ax = plt.subplots()
x_dec.show(ctx=ax)
ax.imshow(cam_map.detach().cpu(), alpha=0.6, extent=(0,224,224,0),
              interpolation='bilinear', cmap='magma');
```

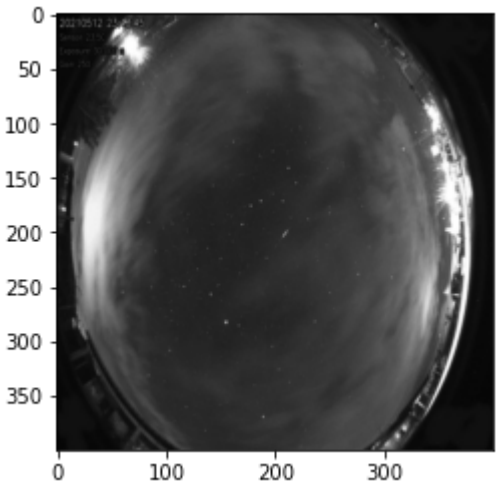


In [99]:
```
learn.predict(img)
```

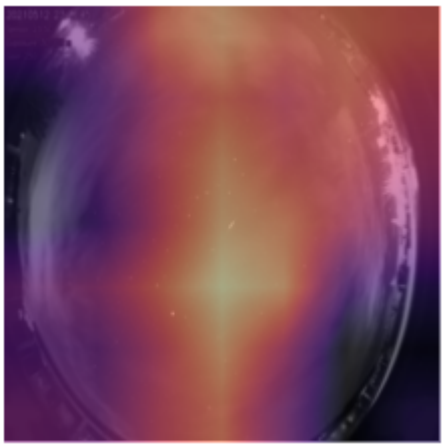
Out[99]:
```
('no-meteor', tensor(1), tensor([0.3189, 0.6811]))
```

In [100]:
```
img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/eliminat_fondo/positive_18.jpg')
x, = first(dls.test_dl([img]))
plt.imshow(img)
```

Out[100]:
```
<matplotlib.image.AxesImage at 0x1e48aed14c0>
```



In [101]:
```
cls = 0
with HookBwd(learn.model[0]) as hookg:
    with Hook(learn.model[0]) as hook:
        output = learn.model.eval()(x.cuda())
        act = hook.stored
    output[0,cls].backward()
    grad = hookg.stored
w = grad[0].mean(dim=[1,2], keepdim=True)
cam_map = (w * act[0]).sum(0)
x_dec = TensorImage(dls.train.decode((x,))[0][0])
_,ax = plt.subplots()
x_dec.show(ctx=ax)
ax.imshow(cam_map.detach().cpu(), alpha=0.6, extent=(0,224,224,0),
              interpolation='bilinear', cmap='magma');
```

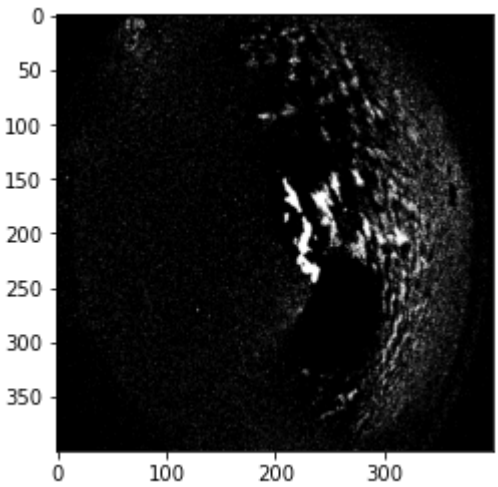


In [102]:
```
learn.predict(img)
```

Out[102]:
```
('meteor', tensor(0), tensor([0.7870, 0.2130]))
```

In [103]:
```
img = PILImage.create('C:/Development/meteor_detector/dataset/v7_adjusted/dataset/test_set/no-meteor/image-20210423001200.jpg')
x, = first(dls.test_dl([img]))
plt.imshow(img)
```

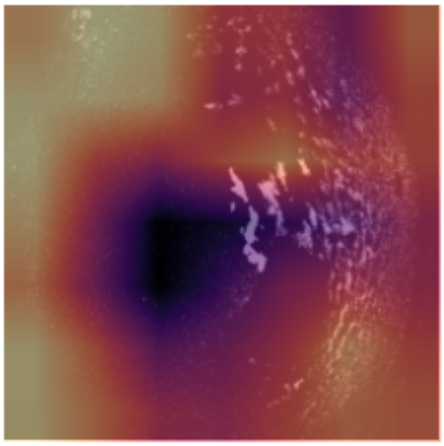
Out[103]:
```
<matplotlib.image.AxesImage at 0x1e46ca7be80>
```



In [104]:
```
learn.predict(img)
```

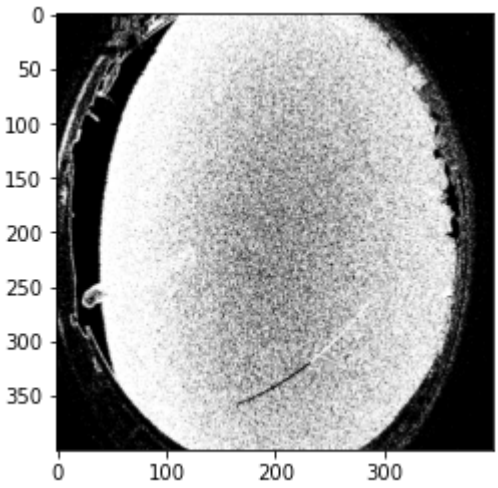
Out[104]:
```
('no-meteor', tensor(1), tensor([0.3139, 0.6861]))
```

In [105]:
```
cls = 1
with HookBwd(learn.model[0]) as hookg:
    with Hook(learn.model[0]) as hook:
        output = learn.model.eval()(x.cuda())
        act = hook.stored
    output[0,cls].backward()
    grad = hookg.stored
w = grad[0].mean(dim=[1,2], keepdim=True)
cam_map = (w * act[0]).sum(0)
x_dec = TensorImage(dls.train.decode((x,))[0][0])
_,ax = plt.subplots()
x_dec.show(ctx=ax)
ax.imshow(cam_map.detach().cpu(), alpha=0.6, extent=(0,224,224,0),
              interpolation='bilinear', cmap='magma');
```



In [106]:
```
img = PILImage.create('C:/Development/meteor_detector/dataset/v7/dataset/test_set/meteor/image-20210321060030.jpg')
x, = first(dls.test_dl([img]))
plt.imshow(img)
```

Out[106]:
```
<matplotlib.image.AxesImage at 0x1e4025477f0>
```



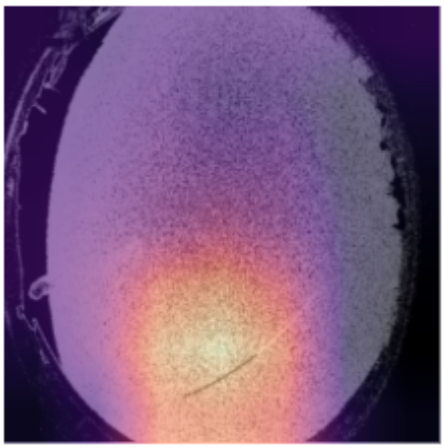
In [107]:
```
learn.predict(img)
```

Out[107]:
```
('meteor', tensor(0), tensor([0.9869, 0.0131]))
```

In [108]:
```
cls = 0
with HookBwd(learn.model[0][-1]) as hookg:
    with Hook(learn.model[0][-1]) as hook:
        output = learn.model.eval()(x.cuda())
        act = hook.stored
    output[0,cls].backward()
    grad = hookg.stored
```

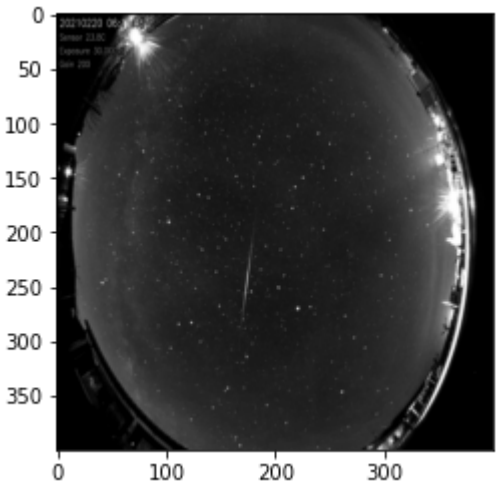
In [109]:
```
w = grad[0].mean(dim=[1,2], keepdim=True)
cam_map = (w * act[0]).sum(0)
```

In [110]:
```
x_dec = TensorImage(dls.train.decode((x,))[0][0])
_,ax = plt.subplots()
x_dec.show(ctx=ax)
ax.imshow(cam_map.detach().cpu(), alpha=0.6, extent=(0,224,224,0),
              interpolation='bilinear', cmap='magma');
```



In [111]:
```
img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/eliminat_fondo/positive_5.jpg')
x, = first(dls.test_dl([img]))
plt.imshow(img)
```

Out[111]:
```
<matplotlib.image.AxesImage at 0x1e4020b4340>
```



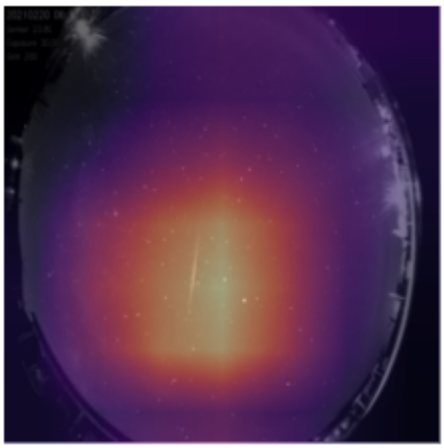
In [112]:
```
learn.predict(img)
```

Out[112]:
```
('meteor', tensor(0), tensor([9.9985e-01, 1.4669e-04]))
```

In [113]:
```
cls = 0
with HookBwd(learn.model[0][-1]) as hookg:
    with Hook(learn.model[0][-1]) as hook:
        output = learn.model.eval()(x.cuda())
        act = hook.stored
    output[0,cls].backward()
    grad = hookg.stored
```

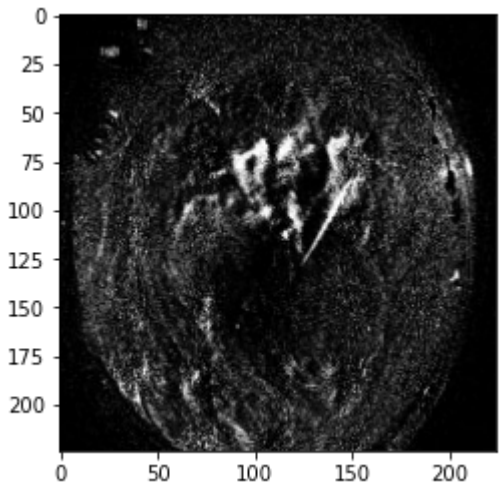
In [114]:
```
w = grad[0].mean(dim=[1,2], keepdim=True)
cam_map = (w * act[0]).sum(0)
```

In [115]:
```
x_dec = TensorImage(dls.train.decode((x,))[0][0])
_,ax = plt.subplots()
x_dec.show(ctx=ax)
ax.imshow(cam_map.detach().cpu(), alpha=0.6, extent=(0,224,224,0),
              interpolation='bilinear', cmap='magma');
```



In [116]:
```
#Detection during pilot in Pujalt
img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/pilot_detection.jpg')
x, = first(dls.test_dl([img]))
plt.imshow(img)
```

Out[116]:
```
<matplotlib.image.AxesImage at 0x1e48a1c4100>
```



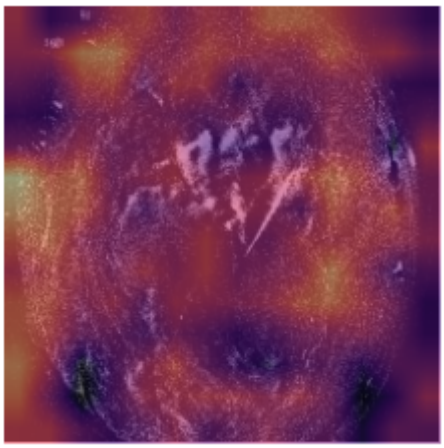
In [117]:
```
learn.predict(img)
```

Out[117]:
```
('meteor', tensor(0), tensor([0.7496, 0.2504]))
```

In [118]:
```
cls = 0
with HookBwd(learn.model[0][-2]) as hookg:
    with Hook(learn.model[0][-2]) as hook:
        output = learn.model.eval()(x.cuda())
        act = hook.stored
    output[0,cls].backward()
    grad = hookg.stored
```

In [119]:
```
w = grad[0].mean(dim=[1,2], keepdim=True)
cam_map = (w * act[0]).sum(0)
```

In [120]:
```
x_dec = TensorImage(dls.train.decode((x,))[0][0])
_,ax = plt.subplots()
x_dec.show(ctx=ax)
ax.imshow(cam_map.detach().cpu(), alpha=0.6, extent=(0,224,224,0),
              interpolation='bilinear', cmap='magma');
```



In [122]:
```
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)
      )
    )
    (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)
      )
    )
    (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)
      )
    )
    (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)
      )
    )
  )
  (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 [ ]: