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 0x000001E32B57E610>
NVIDIA GeForce GTX 1650
```

In [3]:

```
################################DATASET###################
```

In [4]:

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

Out[4]:

```
(#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 [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]:
```
# guardar a fitxer
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]:
```
#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")].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])
df.groupby(["dataset","meteor"]).size()
```

Out[9]:
```
dataset       meteor
test_set      meteor         300
              no-meteor     1000
training_set  meteor        1567
              no-meteor     1567
dtype: int64
```

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 [11]:

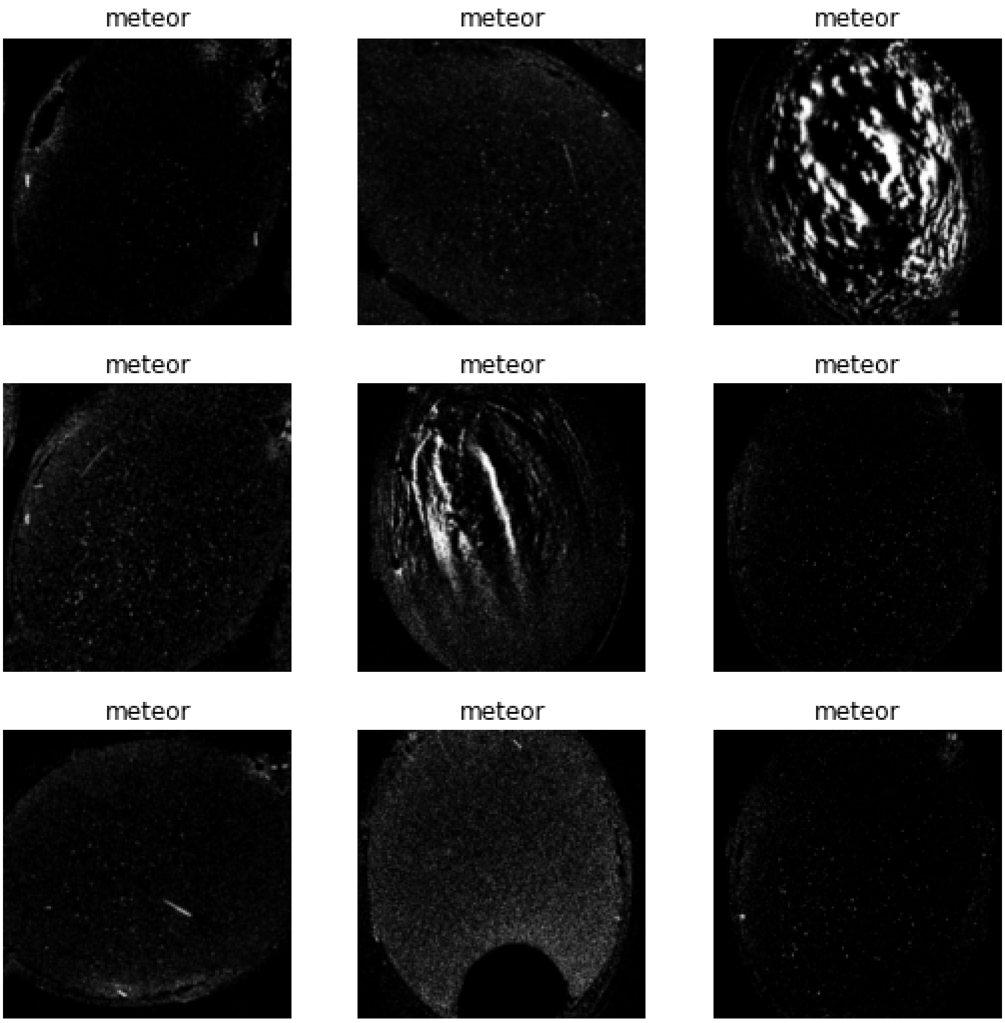
```
dls = ImageDataLoaders.from_df(df[df["dataset"]!="test_set"],
                               folder=origin ,
                               bs=32,
                               batch_tfms=aug_transforms(max_rotate=180,max_warp=0,ma
x_zoom=0), #,pad_mode='zeros'),
                               #batch_tfms=aug_transforms(),
                               item_tfms=[Resize(224)],
                               fn_col=0,
                               label_col=2,
                               shuffle_train=True,
                               drop_last=True,
                               valid_pct=0.2,
                               num_workers=0)
```

In [12]:

```
#aug_transforms?
```

In [13]:

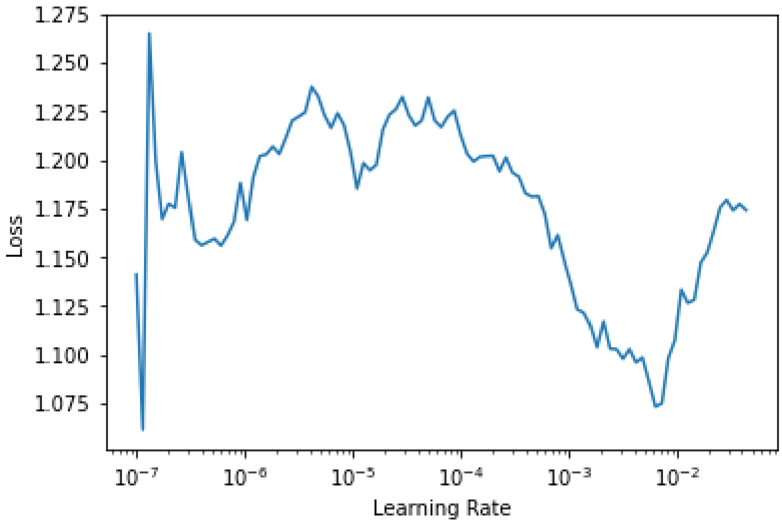
```
dls.show_batch()
```



In [14]:
```
learn = cnn_learner(dls, resnet34,
                metrics=[error_rate, accuracy, Precision(average='macro'), F1Score(
average='macro')],
                    cbs=[EarlyStoppingCallback(patience=25), ShowGraphCallback()]
                   )
```

In [15]:
```
learn.lr_find(end_lr=0.1)
```

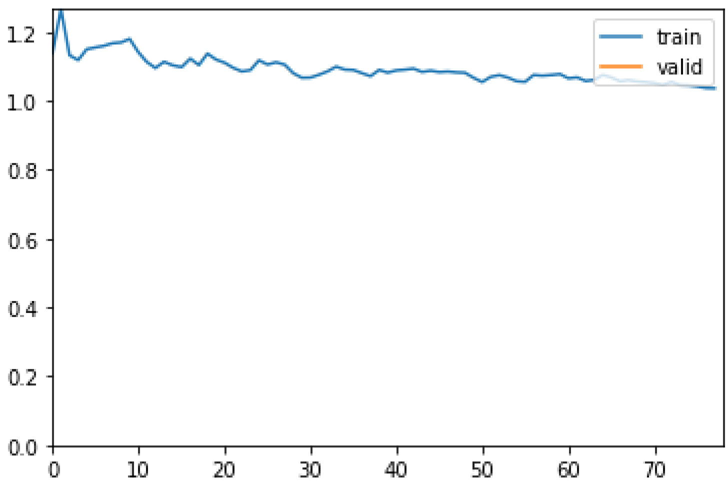
Out[15]:
```
SuggestedLRs(lr_min=0.0006309573538601399, lr_steep=9.12010818865383e-07)
```



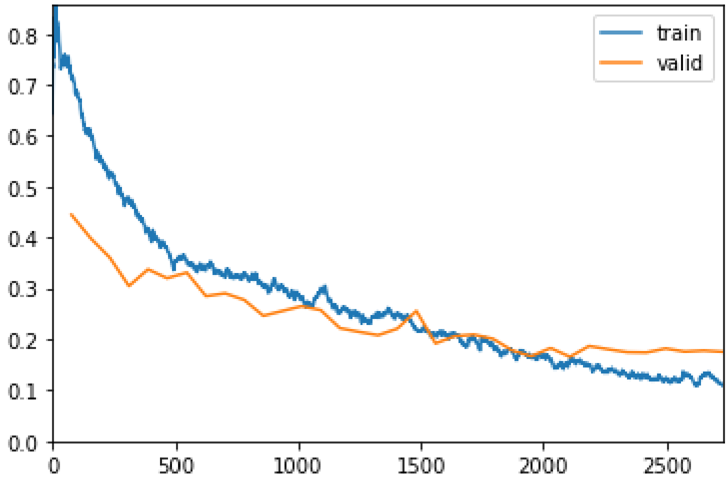
In [16]:

```
learn.fine_tune(35,3*1e-3)
```

epoch	train_loss	valid_loss	error_rate	accuracy	precision_score	f1_score	time
0	1.035894	0.655713	0.255591	0.744409	0.746077	0.743739	00:42



epoch	train_loss	valid_loss	error_rate	accuracy	precision_score	f1_score	time
0	0.714231	0.444611	0.215655	0.784345	0.787071	0.784001	00:55
1	0.601025	0.398551	0.198083	0.801917	0.805621	0.801105	00:54
2	0.527630	0.360468	0.169329	0.830671	0.834315	0.830045	00:55
3	0.476098	0.304456	0.124601	0.875399	0.875388	0.875394	00:54
4	0.416477	0.337251	0.140575	0.859425	0.859593	0.859423	00:55
5	0.378760	0.319829	0.146965	0.853035	0.855687	0.852853	00:54
6	0.353319	0.330873	0.134185	0.865815	0.878339	0.864882	00:53
7	0.334949	0.284512	0.119808	0.880192	0.882418	0.880081	00:55
8	0.321864	0.290034	0.127796	0.872204	0.880764	0.871627	00:55
9	0.329213	0.277125	0.095847	0.904153	0.904626	0.904145	00:54
10	0.302762	0.245778	0.107029	0.892971	0.893734	0.892949	00:57
11	0.302807	0.255302	0.095847	0.904153	0.905051	0.904129	00:56
12	0.284154	0.264714	0.092652	0.907348	0.909914	0.907254	00:58
13	0.296450	0.256983	0.102236	0.897764	0.898087	0.897712	00:59
14	0.253098	0.221467	0.079872	0.920128	0.920479	0.920088	00:56
15	0.250970	0.214154	0.083067	0.916933	0.917454	0.916879	01:00
16	0.247653	0.207563	0.075080	0.924920	0.925987	0.924897	00:59
17	0.248974	0.220460	0.079872	0.920128	0.922351	0.920062	01:00
18	0.218404	0.255301	0.083067	0.916933	0.919141	0.916864	01:01
19	0.208020	0.191421	0.065495	0.934505	0.935328	0.934491	01:01
20	0.212502	0.205709	0.075080	0.924920	0.925508	0.924911	01:02
21	0.202278	0.208797	0.079872	0.920128	0.921053	0.920107	01:00
22	0.186256	0.200574	0.073482	0.926518	0.926624	0.926499	01:00
23	0.176874	0.177722	0.065495	0.934505	0.934505	0.934503	01:00
24	0.165712	0.166215	0.057508	0.942492	0.942821	0.942490	01:01
25	0.157288	0.182090	0.059105	0.940895	0.940943	0.940894	00:58
26	0.161435	0.165068	0.062300	0.937700	0.937710	0.937692	00:57
27	0.148797	0.186211	0.062300	0.937700	0.937952	0.937698	00:58
28	0.140098	0.179922	0.052716	0.947284	0.947284	0.947283	00:58
29	0.130737	0.174124	0.054313	0.945687	0.945707	0.945686	00:57
30	0.128349	0.173539	0.060703	0.939297	0.939624	0.939295	00:59
31	0.124146	0.181350	0.059105	0.940895	0.941028	0.940894	01:00
32	0.129066	0.175320	0.057508	0.942492	0.942579	0.942492	01:00
33	0.132317	0.177232	0.057508	0.942492	0.942579	0.942492	01:01
34	0.109418	0.174929	0.054313	0.945687	0.945676	0.945685	00:59

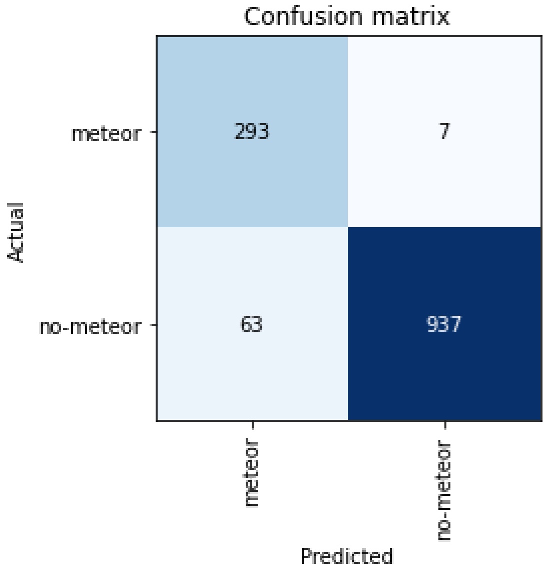


In [17]:
```python
learn.export('C:/Development/meteor_detector/dataset/v7_adjusted/model_test_nou_resnet34.pkl')
#learn = load_learner('C:/Development/meteor_detector/dataset/v7/model2.pkl')
```

In [18]:
```python
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       0.82      0.98      0.89       300
   no-meteor       0.99      0.94      0.96      1000

    accuracy                           0.95      1300
   macro avg       0.91      0.96      0.93      1300
weighted avg       0.95      0.95      0.95      1300
```

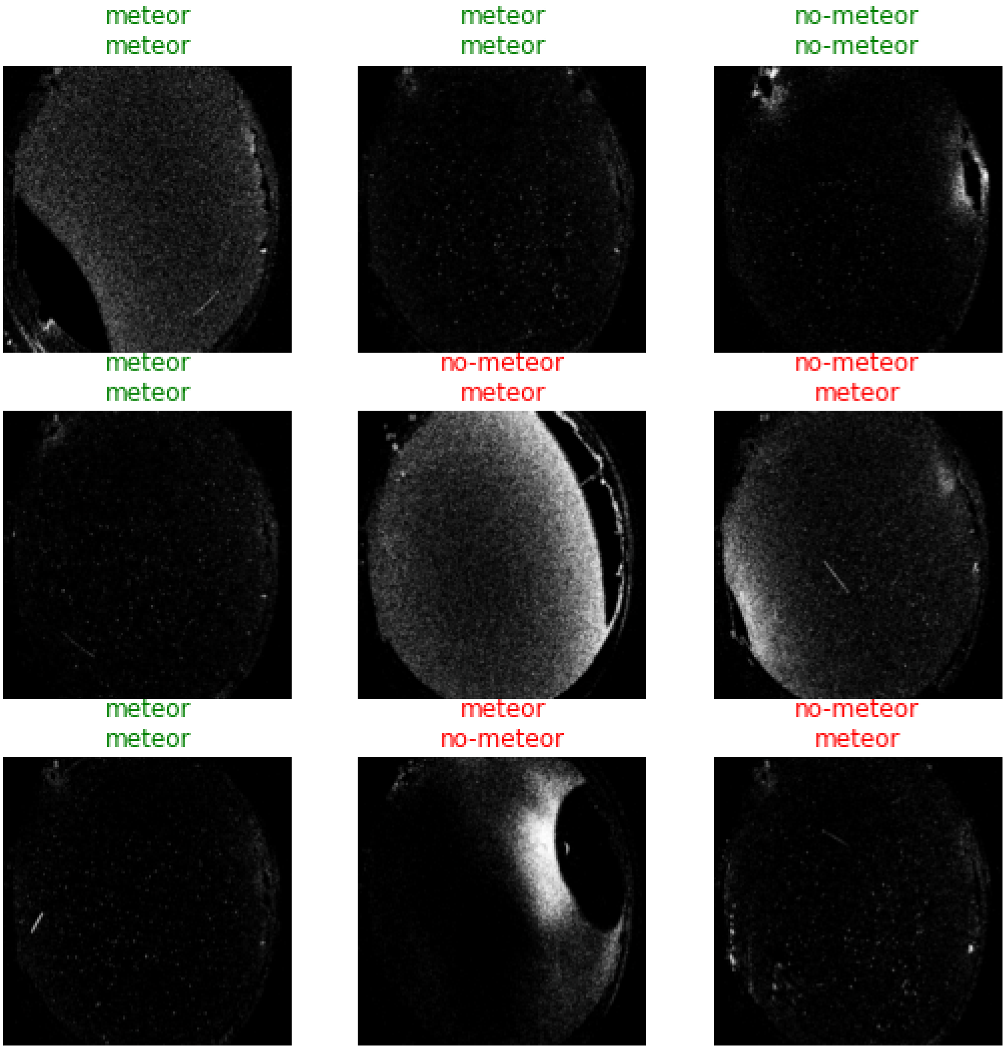
In [19]:
```python
interp.plot_confusion_matrix(figsize=(4, 4))
```



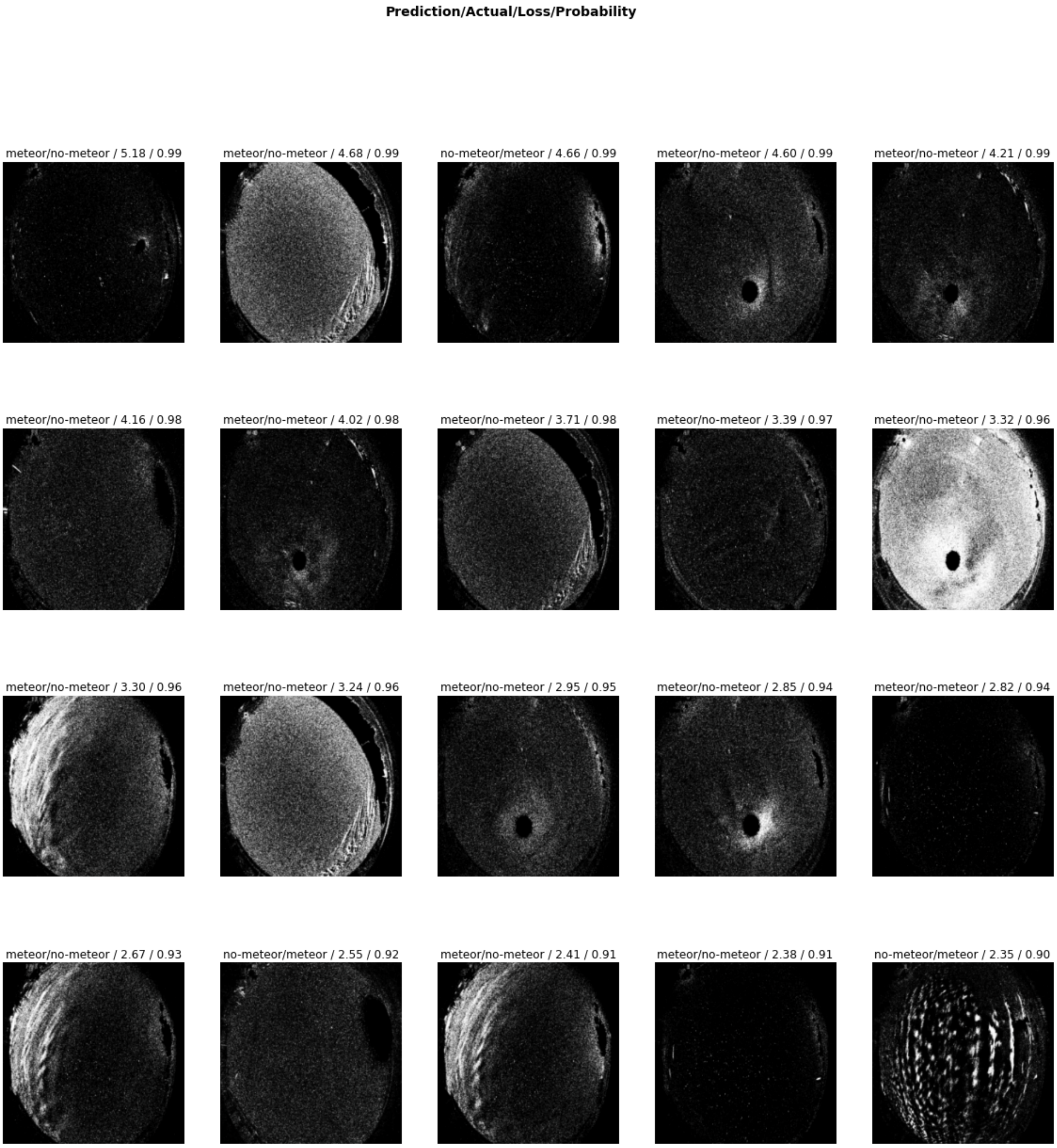
```
In [20]: interp.print_classification_report()
```

```
              precision    recall  f1-score   support

      meteor       0.82      0.98      0.89       300
   no-meteor       0.99      0.94      0.96      1000

    accuracy                           0.95      1300
   macro avg       0.91      0.96      0.93      1300
weighted avg       0.95      0.95      0.95      1300
```

```
In [21]: learn.show_results()
```



```
In [21]: interp.plot_top_losses(k=20,figsize=(20,20))
```



Prediction/Actual/Loss/Probability
meteor/no-meteor / 5.18 / 0.99
meteor/no-meteor / 4.68 / 0.99
no-meteor/meteor / 4.66 / 0.99
meteor/no-meteor / 4.60 / 0.99
meteor/no-meteor / 4.21 / 0.99
meteor/no-meteor / 4.16 / 0.98
meteor/no-meteor / 4.02 / 0.98
meteor/no-meteor / 3.71 / 0.98
meteor/no-meteor / 3.39 / 0.97
meteor/no-meteor / 3.32 / 0.96
meteor/no-meteor / 3.30 / 0.96
meteor/no-meteor / 3.24 / 0.96
meteor/no-meteor / 2.95 / 0.95
meteor/no-meteor / 2.85 / 0.94
meteor/no-meteor / 2.82 / 0.94
meteor/no-meteor / 2.67 / 0.93
no-meteor/meteor / 2.55 / 0.92
meteor/no-meteor / 2.41 / 0.91
meteor/no-meteor / 2.38 / 0.91
no-meteor/meteor / 2.35 / 0.90

In [22]:

```
losses,idxs = interp.top_losses(30)
df[df["dataset"]=="test_set"].iloc[idxs]
#for p in idxs:
#    print(df.iloc[p])
```

Out[22]:

	fn	dataset	meteor
4665	\test_set\no-meteor\image-20210505004757.jpg	test_set	no-meteor
110	\test_set\meteor\image-20210419055134.jpg	test_set	meteor
4791	\test_set\no-meteor\image-20210505015835.jpg	test_set	no-meteor
1519	\test_set\no-meteor\image-20210422215737.jpg	test_set	no-meteor
583	\test_set\no-meteor\image-20210418232222.jpg	test_set	no-meteor
2751	\test_set\no-meteor\image-20210423050439.jpg	test_set	no-meteor
1610	\test_set\no-meteor\image-20210422222805.jpg	test_set	no-meteor
4836	\test_set\no-meteor\image-20210505022107.jpg	test_set	no-meteor
4632	\test_set\no-meteor\image-20210505003055.jpg	test_set	no-meteor
2735	\test_set\no-meteor\image-20210423045638.jpg	test_set	no-meteor
1223	\test_set\no-meteor\image-20210419045958.jpg	test_set	no-meteor
4699	\test_set\no-meteor\image-20210505011200.jpg	test_set	no-meteor
1150	\test_set\no-meteor\image-20210419041153.jpg	test_set	no-meteor
4567	\test_set\no-meteor\image-20210504234951.jpg	test_set	no-meteor
1168	\test_set\no-meteor\image-20210419042224.jpg	test_set	no-meteor
4557	\test_set\no-meteor\image-20210504233950.jpg	test_set	no-meteor
124	\test_set\meteor\image-20210423044839.jpg	test_set	meteor
4670	\test_set\no-meteor\image-20210505005027.jpg	test_set	no-meteor
4792	\test_set\no-meteor\image-20210505015905.jpg	test_set	no-meteor
1518	\test_set\no-meteor\image-20210422215717.jpg	test_set	no-meteor
1299	\test_set\no-meteor\image-20210419054733.jpg	test_set	no-meteor
4695	\test_set\no-meteor\image-20210505010959.jpg	test_set	no-meteor
4662	\test_set\no-meteor\image-20210505004627.jpg	test_set	no-meteor
4955	\test_set\no-meteor\image-20210505032615.jpg	test_set	no-meteor
4832	\test_set\no-meteor\image-20210505021907.jpg	test_set	no-meteor
1232	\test_set\no-meteor\image-20210419050429.jpg	test_set	no-meteor
2856	\test_set\no-meteor\image-20210423055913.jpg	test_set	no-meteor
4740	\test_set\no-meteor\image-20210505013302.jpg	test_set	no-meteor
1495	\test_set\no-meteor\image-20210422214842.jpg	test_set	no-meteor
2651	\test_set\no-meteor\image-20210423041802.jpg	test_set	no-meteor

In [23]:

```
img = PILImage.create('C:/Development/meteor_detector/dataset/v7_adjusted/dataset/test_set/no-meteor/image-20210505004827.jpg')
```

In [24]:

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

Out[24]:

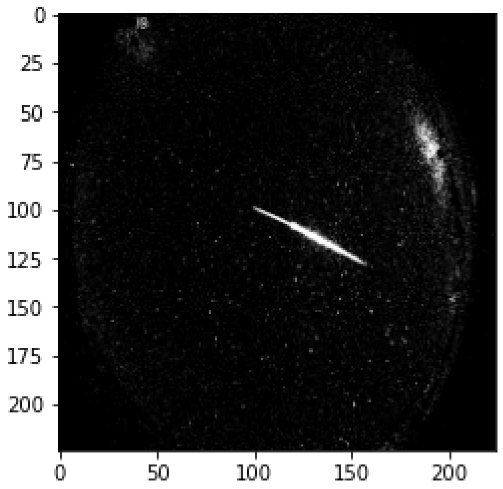
```
('meteor', tensor(0), tensor([0.8797, 0.1203]))
```

In [25]:

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

Out[25]:

```
<matplotlib.image.AxesImage at 0x1e303c6e1f0>
```



In [26]:

```
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 [27]:

```
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 [32]:

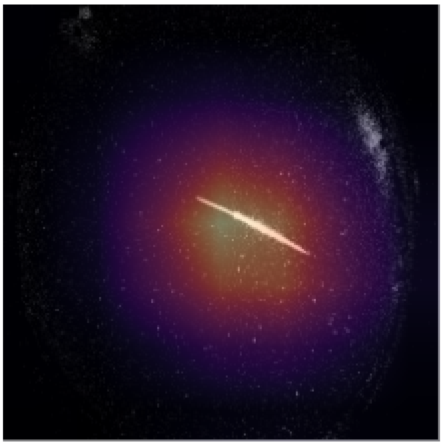
```
cls = 0
with HookBwd(learn.model[0]) as hookg:
    with Hook(learn.model[0]) as hook:
        output = learn.model.eval()(x.cuda())
        act = hook.stored
    output[0,cls].backward()
    grad = hookg.stored
```

In [33]:

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

In [34]:

```
x_dec = TensorImage(dls.train.decode((x,))[0][0])
_,ax = plt.subplots()
x_dec.show(ctx=ax)
ax.imshow(cam_map.detach().cpu(), alpha=0.5, extent=(0,224,224,0),
              interpolation='bilinear', cmap='magma');
```



In [35]:

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

Out[35]:

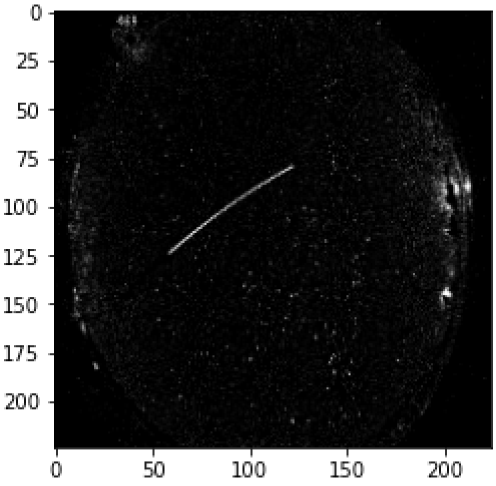
```
('meteor', tensor(0), tensor([0.9945, 0.0055]))
```

In [51]:

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

Out[51]:

```
<matplotlib.image.AxesImage at 0x1e316f04760>
```



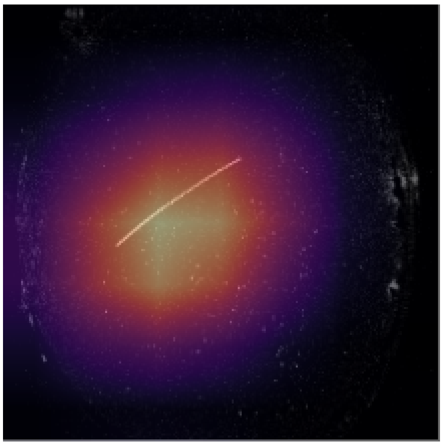
In [52]:

```
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 [53]:

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

In [54]:
```
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');
```

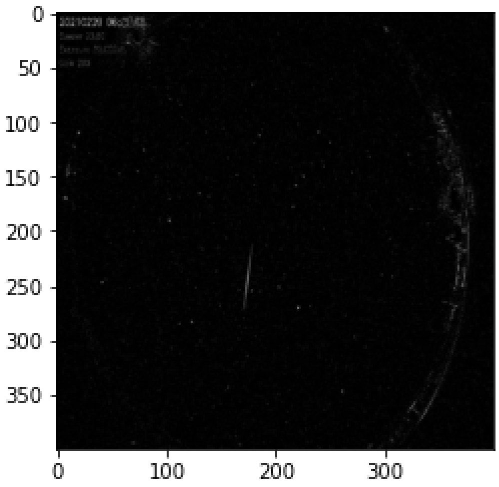


In [55]:
```
learn.predict(img)
```

Out[55]:
```
('meteor', tensor(0), tensor([9.9998e-01, 1.9137e-05]))
```

In [59]:
```
img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/eliminat_fondo/positive_1.jpg')
x, = first(dls.test_dl([img]))
plt.imshow(img)
```

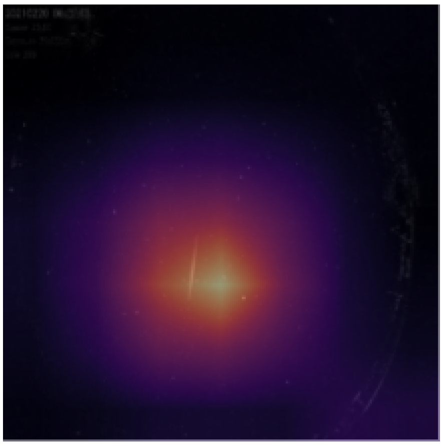
Out[59]:
```
<matplotlib.image.AxesImage at 0x1e30b82b4f0>
```



In [63]:
```
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 [64]:
```
w = grad[0].mean(dim=[1,2], keepdim=True)
cam_map = (w * act[0]).sum(0)
```

In [65]:
```
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');
```

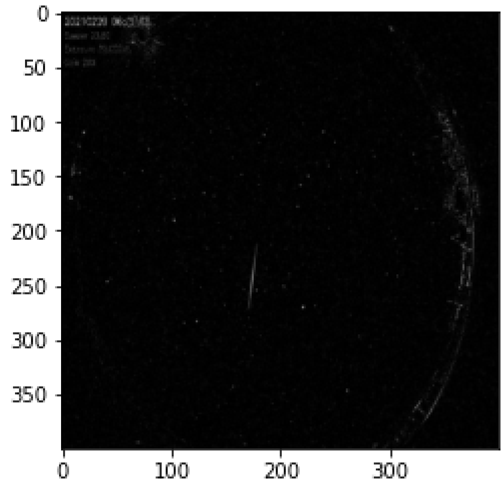


In [67]:
```
learn.predict(img)
```

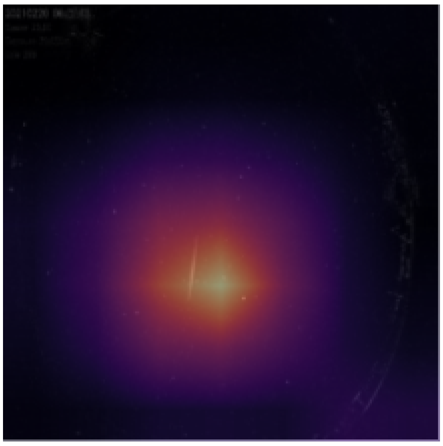
Out[67]:
```
('meteor', tensor(0), tensor([9.9986e-01, 1.3943e-04]))
```

In [68]:
```
img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/eliminat_fondo/positive_1.jpg')
x, = first(dls.test_dl([img]))
plt.imshow(img)
```

Out[68]:
```
<matplotlib.image.AxesImage at 0x1e300078dc0>
```

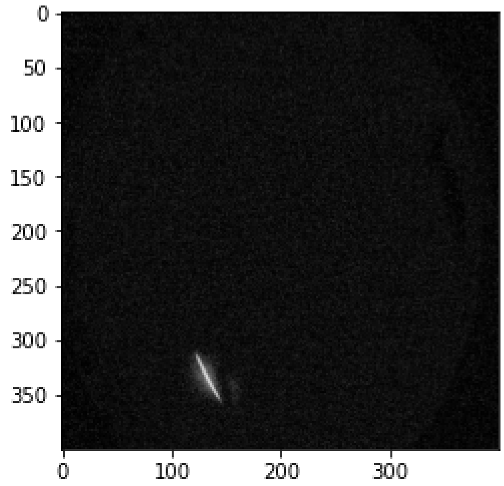


```
In [69]: cls = 0
         with HookBwd(learn.model[0]) as hookg:
             with Hook(learn.model[0]) as hook:
                 output = learn.model.eval()(x.cuda())
                 act = hook.stored
             output[0,cls].backward()
             grad = hookg.stored
         w = grad[0].mean(dim=[1,2], 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');
```

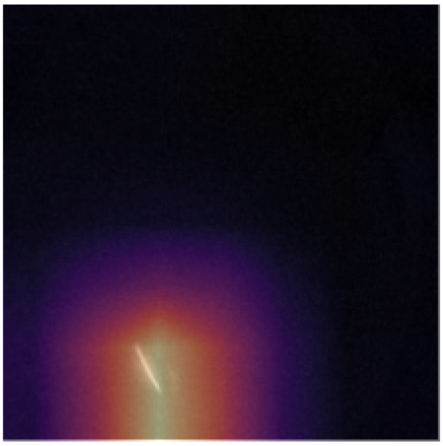


```
In [70]: learn.predict(img)

Out[70]: ('meteor', tensor(0), tensor([9.9986e-01, 1.3943e-04]))
```

```
In [71]: img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/eliminat_fondo/positive_3.jpg')
         x, = first(dls.test_dl([img]))
         plt.imshow(img)

Out[71]: <matplotlib.image.AxesImage at 0x1e3084c1b20>
```



In [72]:
```
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');
```

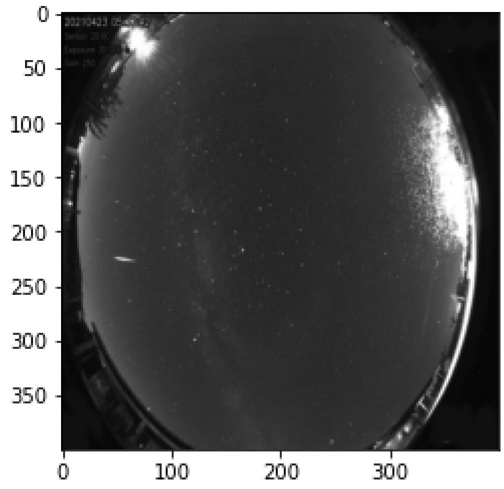


In [73]:
```
learn.predict(img)
```

Out[73]:
```
('meteor', tensor(0), tensor([0.9183, 0.0817]))
```

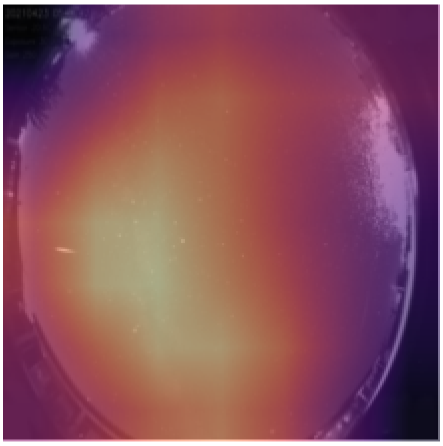
In [78]:
```
img = PILImage.create('C:/Development/meteor_detector/dataset/Positius/eliminat_fondo/positive_14.jpg')
x, = first(dls.test_dl([img]))
plt.imshow(img)
```

Out[78]:
```
<matplotlib.image.AxesImage at 0x1e33bc530d0>
```



In [79]:

```
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 [80]:

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

Out[80]:

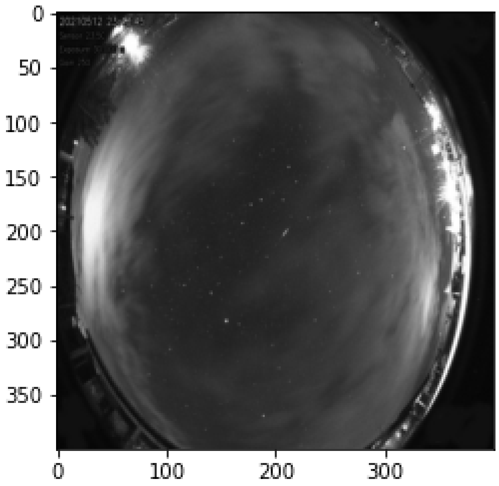
```
('meteor', tensor(0), tensor([9.9974e-01, 2.6009e-04]))
```

In [81]:

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

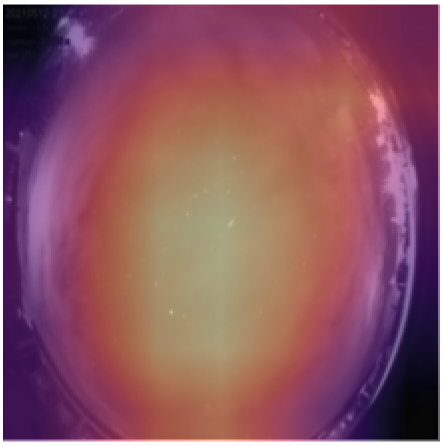
Out[81]:

```
<matplotlib.image.AxesImage at 0x1e33bcfbb80>
```



In [82]:

```
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');
```

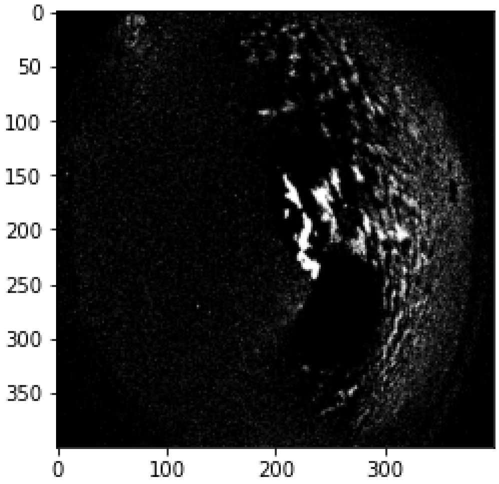


In [84]:

```
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[84]:

```
<matplotlib.image.AxesImage at 0x1e343152670>
```



In [85]:

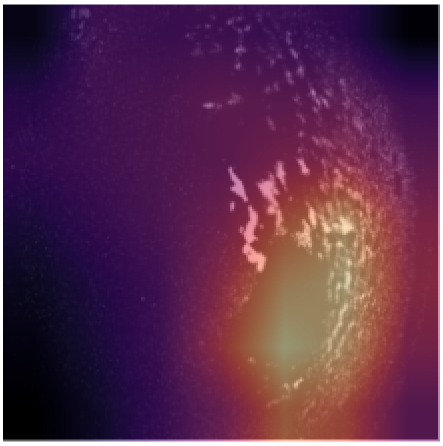
```
learn.predict(img)
```

Out[85]:

```
('no-meteor', tensor(1), tensor([1.5450e-04, 9.9985e-01]))
```

In [87]:

```
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');
```

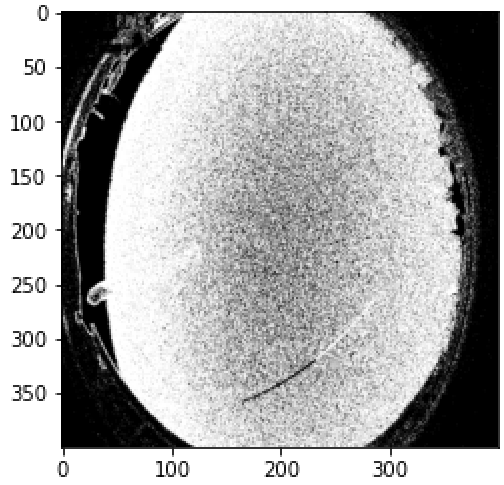


In [46]:

```
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[46]:

```
<matplotlib.image.AxesImage at 0x1e303ca6c40>
```



In [47]:

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

Out[47]:

```
('meteor', tensor(0), tensor([9.9968e-01, 3.1693e-04]))
```

In [48]:

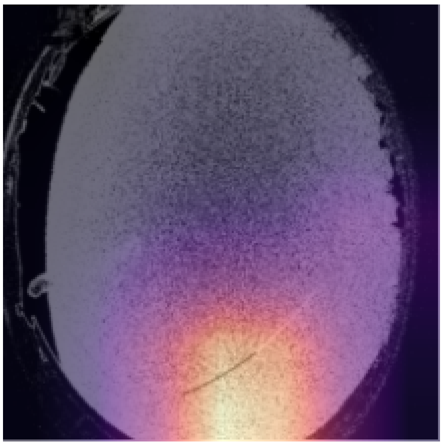
```
cls = 0
with HookBwd(learn.model[0][-1]) as hookg:
    with Hook(learn.model[0][-1]) as hook:
        output = learn.model.eval()(x.cuda())
        act = hook.stored
    output[0,cls].backward()
    grad = hookg.stored
```

In [49]:

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

In [50]:

```
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 [ ]: