In [43]:

```
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow.keras as keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import MobileNet
```

In [44]:

```
dataset_dir = r"C:\Users\arnav\Desktop\Me\UST Global\Dataset\DRIVE"
```

In [45]:

```
image_size = (224, 224)
batch_size = 32
```

In [46]:

```
data_generator = ImageDataGenerator(
    rescale=1.0 / 255.0,
    rotation_range=20,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True,
    validation_split=0.2
)
```

In [47]:

```
train_generator = data_generator.flow_from_directory(
    dataset_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical',
    subset='training'
)
```

```
Found 48 images belonging to 2 classes.
```

In [48]:

```
validation_generator = data_generator.flow_from_directory(
    dataset_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical',
    subset='validation'
)
```

```
Found 12 images belonging to 2 classes.
```

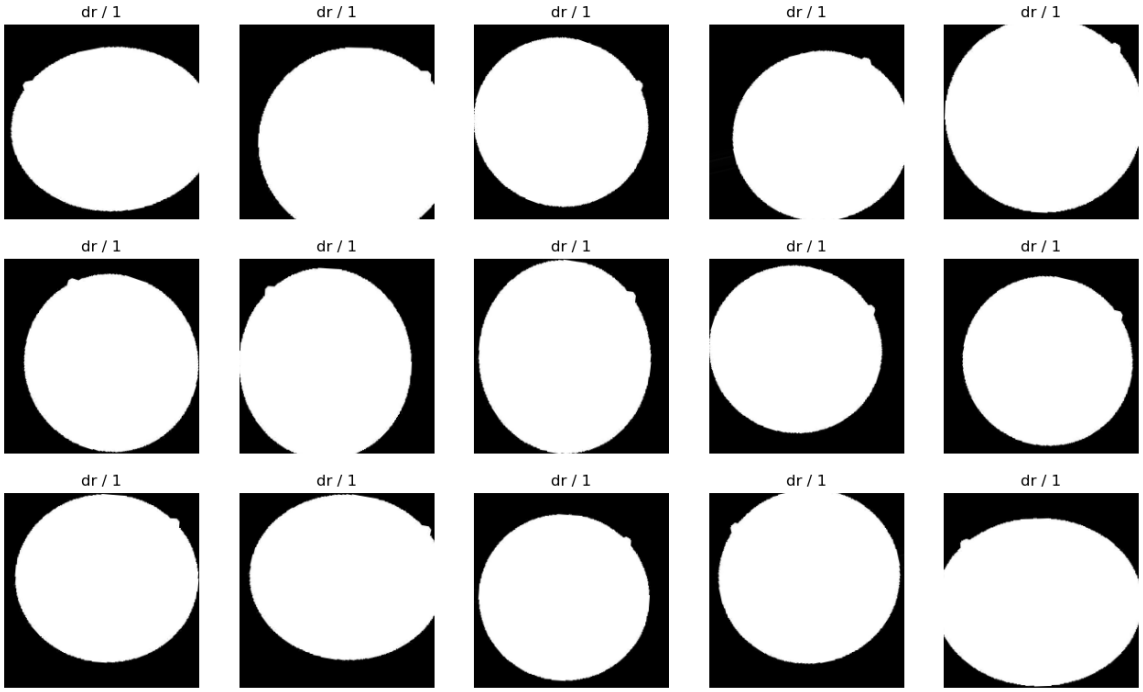
In [49]:

```
class_labels = {
    0: 'no dr',
    1: 'dr'
}
```

In [50]:

```
plt.figure(figsize=(16, 16))
j = 1
for i in np.random.randint(0, len(train_generator), 15):
    plt.subplot(5, 5, j)
    j += 1
    plt.imshow(train_generator[i][0][0], cmap="Greys")
    plt.axis('off')
    label = np.argmax(train_generator[i][1][0])
    plt.title('{} / {}'.format(class_labels[label], label))

plt.show()
```



In [51]:

```
base_model = MobileNet(include_top=False, input_shape=(224, 224, 3))
base_model.trainable = False
```

In [52]:

```
model = keras.models.Sequential([
    base_model,
    keras.layers.GlobalAveragePooling2D(),
    keras.layers.Dense(256, activation='relu'),
    keras.layers.Dense(2, activation='softmax')
])
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 mobilenet_1.00_224 (Functio  (None, 7, 7, 1024)       3228864
 nal)

 global_average_pooling2d_2   (None, 1024)             0
 (GlobalAveragePooling2D)

 dense_4 (Dense)             (None, 256)               262400

 dense_5 (Dense)             (None, 2)                 514

=================================================================
Total params: 3,491,778
Trainable params: 262,914
Non-trainable params: 3,228,864
_________________________________________________________________
```

In [53]:

```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

In [54]:

```
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // batch_size,
    epochs=20,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // batch_size
)
```

```
Epoch 1/20
1/1 [==============================] - 3s 3s/step - loss: 1.3137 - accurac
y: 0.5000
Epoch 2/20
1/1 [==============================] - 1s 1s/step - loss: 0.8892 - accurac
y: 0.7188
Epoch 3/20
1/1 [==============================] - 1s 1s/step - loss: 0.7201 - accurac
y: 0.7188
Epoch 4/20
1/1 [==============================] - 1s 682ms/step - loss: 0.2865 - accu
racy: 0.8750
Epoch 5/20
1/1 [==============================] - 1s 991ms/step - loss: 0.4687 - accu
racy: 0.8438
Epoch 6/20
1/1 [==============================] - 1s 583ms/step - loss: 0.8505 - accu
racy: 0.6875
Epoch 7/20
1/1 [==============================] - 1s 1s/step - loss: 0.3397 - accurac
y: 0.7812
Epoch 8/20
1/1 [==============================] - 1s 1s/step - loss: 0.3100 - accurac
y: 0.8438
Epoch 9/20
1/1 [==============================] - 1s 645ms/step - loss: 0.4161 - accu
racy: 0.7500
Epoch 10/20
1/1 [==============================] - 1s 1s/step - loss: 0.4001 - accurac
y: 0.7500
Epoch 11/20
1/1 [==============================] - 1s 1s/step - loss: 0.3040 - accurac
y: 0.9375
Epoch 12/20
1/1 [==============================] - 1s 1s/step - loss: 0.3565 - accurac
y: 0.7812
Epoch 13/20
1/1 [==============================] - 1s 629ms/step - loss: 0.2807 - accu
racy: 0.8750
Epoch 14/20
1/1 [==============================] - 1s 1s/step - loss: 0.1983 - accurac
y: 0.9375
Epoch 15/20
1/1 [==============================] - 1s 675ms/step - loss: 0.4213 - accu
racy: 0.8125
Epoch 16/20
1/1 [==============================] - 1s 695ms/step - loss: 0.3384 - accu
racy: 0.8750
Epoch 17/20
1/1 [==============================] - 1s 582ms/step - loss: 0.2019 - accu
racy: 0.8750
Epoch 18/20
1/1 [==============================] - 1s 627ms/step - loss: 0.1473 - accu
racy: 0.9375
Epoch 19/20
1/1 [==============================] - 1s 1s/step - loss: 0.3179 - accurac
y: 0.8125
Epoch 20/20
1/1 [==============================] - 1s 734ms/step - loss: 0.2069 - accu
racy: 0.9375
```

In [55]:

```
test_dir = r"C:\Users\arnav\Desktop\Me\UST Global\Dataset\DRIVE\test"
```

In [56]:

```
test_generator = data_generator.flow_from_directory(
    test_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False
)
```

```
Found 20 images belonging to 2 classes.
```

In [57]:

```
test_loss, test_accuracy = model.evaluate(test_generator, verbose=1)
predictions = model.predict(test_generator)
predicted_labels = np.argmax(predictions, axis=1)
true_labels = test_generator.classes
print('Test loss:', test_loss)
print('Test accuracy:', test_accuracy)
```

```
1/1 [==============================] - 1s 1s/step - loss: 0.4722 - accurac
y: 0.8000
1/1 [==============================] - 1s 1s/step
Test loss: 0.4721725583076477
Test accuracy: 0.800000011920929
```

In [58]:

```
from sklearn.metrics import classification_report
classification_report = classification_report(true_labels, predicted_labels, zero_divisic
print('Classification Report:\n', classification_report)
```

```
Classification Report:
               precision    recall  f1-score   support

           0       1.00      0.90      0.95        20
           1       0.00      1.00      0.00         0

    accuracy                           0.90        20
   macro avg       0.50      0.95      0.47        20
weighted avg       1.00      0.90      0.95        20
```

In [61]:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(true_labels, predicted_labels)
tn, fp, fn, tp = cm[0, 0], cm[0, 1], cm[1, 0], cm[1, 1]

if (tp + fn) != 0:
    sensitivity = tp / (tp + fn)
else:
    sensitivity = 0.0
specificity = tn / (tn + fp)

print("Specificity:", specificity)

print(cm)
```

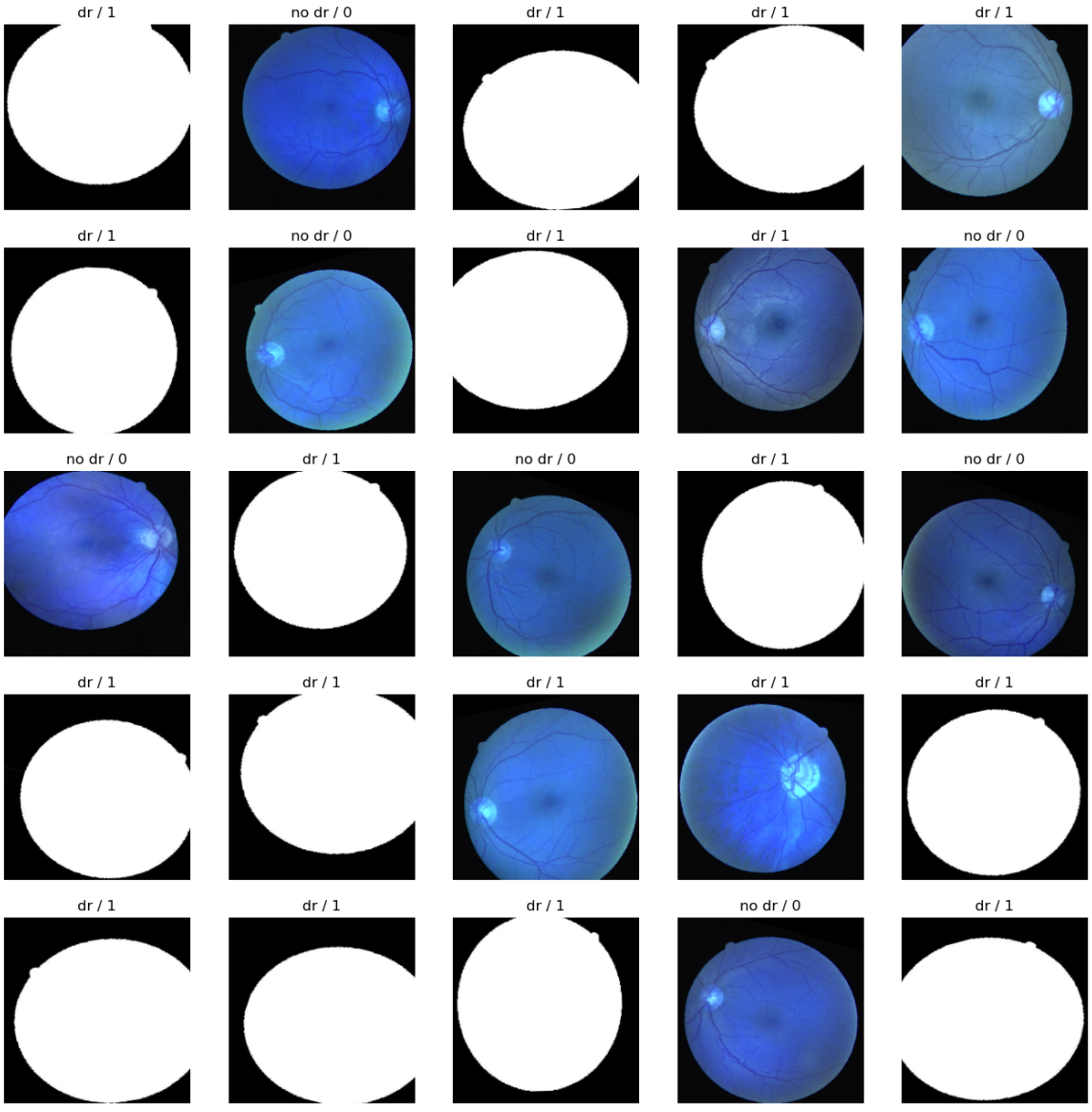
```
Specificity: 0.9
[[18  2]
 [ 0  0]]
```

In [60]:

```
plt.figure(figsize=(16, 16))
j = 1
for batch in train_generator:
    images, labels = batch
    for i in range(len(images)):
        plt.subplot(5, 5, j)
        j += 1
        img = images[i]
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        plt.imshow(img)
        plt.axis('off')
        label = np.argmax(labels[i])
        plt.title('{} / {}'.format(class_labels[label], label))

        if j > 25:
            break
    if j > 25:
        break

plt.show()
```



In [ ]:

```
```