```
import os
import cv2
import numpy as np

def load_and_preprocess_images(image_dir):
    images = []
    for filename in os.listdir(image_dir):
        img = cv2.imread(os.path.join(image_dir, filename))
        img = cv2.resize(img, (224, 224))  # Resize to match model input
        images.append(img)
    return np.array(images)
```

```
from tensorflow.keras import layers, models

def create_model():
    model = models.Sequential([
        layers.Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Conv2D(64, (3, 3), activation='relu'),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Flatten(),
        layers.Dense(64, activation='relu'),
        layers.Dense(1, activation='sigmoid')  # For binary classification
    ])
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    return model
```

```
import numpy as np
from model import create_model
from data_preprocessing import load_and_preprocess_images

# Load and preprocess data
X_train = load_and_preprocess_images('data/images/train/')
y_train = np.load('data/labels/train_labels.npy')  # Load labels

# Create and train model
model = create_model()
model.fit(X_train, y_train, epochs=10, batch_size=32)
```

```
import numpy as np
from model import create_model
from data_preprocessing import load_and_preprocess_images

# Load model
model = create_model()
model.load_weights('model_weights.h5')  # Load trained weights

# Load and preprocess new image
new_image = load_and_preprocess_images('data/images/test/')[0]
predictions = model.predict(np.expand_dims(new_image, axis=0))

print("Predicted probability of cancer:", predictions[0][0])
```