In [1]:
```
import os
import zipfile
```

In [2]:
```
rock_dir = os.path.join('C:/Users/Block-03-EE/Desktop/Machine_Learning/rps/ro
paper_dir = os.path.join('C:/Users/Block-03-EE/Desktop/Machine_Learning/rps/p
scissors_dir = os.path.join('C:/Users/Block-03-EE/Desktop/Machine_Learning/rp

print('total training rock images:', len(os.listdir(rock_dir)))
print('total training paper images:', len(os.listdir(paper_dir)))
print('total training scissors images:', len(os.listdir(scissors_dir)))

rock_files = os.listdir(rock_dir)
print(rock_files[:10])

paper_files = os.listdir(paper_dir)
print(paper_files[:10])

scissors_files = os.listdir(scissors_dir)
print(scissors_files[:10])
```

```
total training rock images: 840
total training paper images: 840
total training scissors images: 840
['rock01-000.png', 'rock01-001.png', 'rock01-002.png', 'rock01-003.png', 'r
ock01-004.png', 'rock01-005.png', 'rock01-006.png', 'rock01-007.png', 'rock
01-008.png', 'rock01-009.png']
['paper01-000.png', 'paper01-001.png', 'paper01-002.png', 'paper01-003.pn
g', 'paper01-004.png', 'paper01-005.png', 'paper01-006.png', 'paper01-007.p
ng', 'paper01-008.png', 'paper01-009.png']
['scissors01-000.png', 'scissors01-001.png', 'scissors01-002.png', 'scissor
s01-003.png', 'scissors01-004.png', 'scissors01-005.png', 'scissors01-006.p
ng', 'scissors01-007.png', 'scissors01-008.png', 'scissors01-009.png']
```

In [3]:

```
%matplotlib inline

import matplotlib.pyplot as plt
import matplotlib.image as mpimg

pic_index = 2

next_rock = [os.path.join(rock_dir, fname)
                for fname in rock_files[pic_index-2:pic_index]]
next_paper = [os.path.join(paper_dir, fname)
                for fname in paper_files[pic_index-2:pic_index]]
next_scissors = [os.path.join(scissors_dir, fname)
                for fname in scissors_files[pic_index-2:pic_index]]

for i, img_path in enumerate(next_rock+next_paper+next_scissors):
  #print(img_path)
  img = mpimg.imread(img_path)
  plt.imshow(img)
  plt.axis('Off')
  plt.show()
```

In [4]:
```
import tensorflow as tf
import keras_preprocessing
from keras_preprocessing import image
from keras_preprocessing.image import ImageDataGenerator

TRAINING_DIR = "C:/Users/Block-03-EE/Desktop/Machine_Learning/rps"
training_datagen = ImageDataGenerator(
      rescale = 1./255,
      rotation_range=40,
      width_shift_range=0.2,
      height_shift_range=0.2,
      shear_range=0.2,
      zoom_range=0.2,
      horizontal_flip=True,
      fill_mode='nearest')

VALIDATION_DIR = "C:/Users/Block-03-EE/Desktop/Machine_Learning/rps-test-set"
validation_datagen = ImageDataGenerator(rescale = 1./255)

train_generator = training_datagen.flow_from_directory(
    TRAINING_DIR,
    target_size=(150,150),
    class_mode='categorical'
)
```

```
Found 2520 images belonging to 3 classes.
```

In [5]:
```
validation_generator = validation_datagen.flow_from_directory(
    VALIDATION_DIR,
    target_size=(150,150),
    class_mode='categorical'
)
```

```
Found 372 images belonging to 3 classes.
```

In [6]:

```
model = tf.keras.models.Sequential([
    # Note the input shape is the desired size of the image 150x150 with 3 by
    # This is the first convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(150, 15
    tf.keras.layers.MaxPooling2D(2, 2),
    # The second convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # The third convolution
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # The fourth convolution
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # Flatten the results to feed into a DNN
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dropout(0.5),
    # 512 neuron hidden layer
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(3, activation='softmax')
])
```

In [7]:

```
model.summary()

model.compile(loss = 'categorical_crossentropy', optimizer='rmsprop', metrics

history = model.fit_generator(train_generator, epochs=10, validation_data = v

model.save("rps.h5")
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 148, 148, 64)      1792
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 74, 74, 64)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 72, 72, 64)        36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 36, 36, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 34, 34, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 17, 17, 128)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 15, 15, 128)       147584
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 7, 7, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 6272)              0
_________________________________________________________________
dropout (Dropout)            (None, 6272)              0
_________________________________________________________________
dense (Dense)                (None, 512)               3211776
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 1539
=================================================================
Total params: 3,473,475
Trainable params: 3,473,475
Non-trainable params: 0
_________________________________________________________________

C:\Users\Block-03-EE\anaconda3\lib\site-packages\tensorflow\python\keras\en
gine\training.py:1844: UserWarning: `Model.fit_generator` is deprecated and
will be removed in a future version. Please use `Model.fit`, which supports
generators.
  warnings.warn('`Model.fit_generator` is deprecated and '

Epoch 1/10
79/79 [==============================] - 80s 1s/step - loss: 1.7915 - acc
uracy: 0.3707 - val_loss: 0.9882 - val_accuracy: 0.6102
Epoch 2/10
79/79 [==============================] - 79s 1s/step - loss: 0.9595 - acc
uracy: 0.5690 - val_loss: 0.4332 - val_accuracy: 0.8710
Epoch 3/10
79/79 [==============================] - 75s 950ms/step - loss: 0.6423 -
```

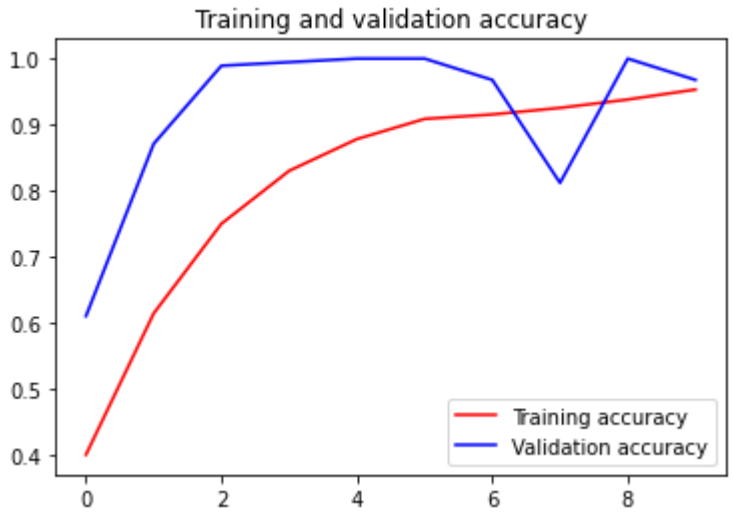
```
accuracy: 0.7177 - val_loss: 0.1615 - val_accuracy: 0.9892
Epoch 4/10
79/79 [==============================] - 76s 967ms/step - loss: 0.4198 -
accuracy: 0.8334 - val_loss: 0.1309 - val_accuracy: 0.9946
Epoch 5/10
79/79 [==============================] - 75s 949ms/step - loss: 0.3288 -
accuracy: 0.8665 - val_loss: 0.0480 - val_accuracy: 1.0000
Epoch 6/10
79/79 [==============================] - 78s 992ms/step - loss: 0.2376 -
accuracy: 0.9122 - val_loss: 0.0400 - val_accuracy: 1.0000
Epoch 7/10
79/79 [==============================] - 75s 943ms/step - loss: 0.2291 -
accuracy: 0.9194 - val_loss: 0.0741 - val_accuracy: 0.9677
Epoch 8/10
79/79 [==============================] - 76s 957ms/step - loss: 0.2355 -
accuracy: 0.9112 - val_loss: 0.4068 - val_accuracy: 0.8118
Epoch 9/10
79/79 [==============================] - 78s 981ms/step - loss: 0.1995 -
accuracy: 0.9334 - val_loss: 0.0275 - val_accuracy: 1.0000
Epoch 10/10
79/79 [==============================] - 78s 982ms/step - loss: 0.1592 -
accuracy: 0.9465 - val_loss: 0.0882 - val_accuracy: 0.9677
```

In [9]:
```
import matplotlib.pyplot as plt
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(acc))

plt.plot(epochs, acc, 'r', label='Training accuracy')
plt.plot(epochs, val_acc, 'b', label='Validation accuracy')
plt.title('Training and validation accuracy')
plt.legend(loc=0)
plt.figure()


plt.show()
```




```
<Figure size 432x288 with 0 Axes>
```

In [ ]:
```

```