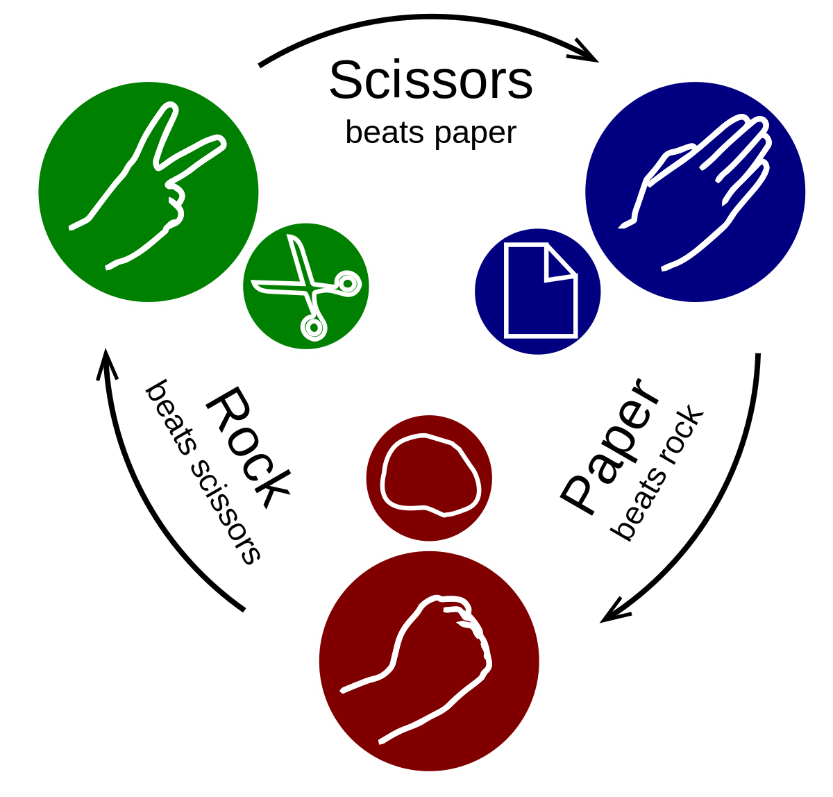
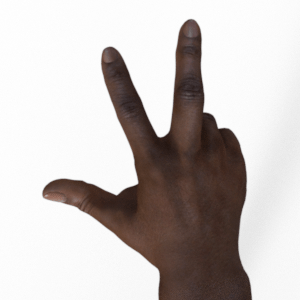
**Classification of Scissors-Rock-Paper Images**

**Description about data:**

***Rock paper scissors*** is a hand game usually played between two people, in which each player simultaneously forms one of three shapes with an outstretched hand. These shapes are "rock" (a closed fist), "paper" (a flat hand), and "scissors" (a fist with the index finger and middle finger extended, forming a V). "Scissors" is identical to the two-fingered V sign (also indicating "victory" or "peace") except that it is pointed horizontally instead of being held upright in the air. The game is as shown in the below given image.



The rps dataset contains images from a variety of different hands, from different races, ages and genders, posed into Rock / Paper or Scissors and labelled as such. Here are a few examples showing some of the poses, and the diversity of hand model used.

Rock Paper Scissors

**Tasks:**

You are given three folders viz. rps-training-set.zip, rps-test-set.zip and rps-validation.zip containing the images of rock, paper and scissors images. You must use them for training, testing and validation respectively.

Your tasks are as listed below:

1. Create appropriate folder structure on disk (or google drive) as per the requirement of Image Generator.
2. Mount the drive and read the data.
3. Visualize some of the images (paper, scissors and rock) from training and testing sets.
4. Build a suitable CNN model with at least 3 convolution layers.
5. Use appropriate activation functions and pooling layers.
6. Compile the model.
7. Train the model and observe training and validation accuracy.
8. Plot the appropriate loss and accuracy.
9. Check whether any overfitting is there.
10. Display confusion matrix.
11. Create a data frame containing two columns viz. name of the class and number of misclassifications.
12. Use Image augmentation techniques to improve the model performance. Do not store the augmented images on the disk.
13. Train the model with batch size of 10 with augmented images.
14. Observe training and validation accuracy.
15. Plot the appropriate loss and accuracy.
16. Display confusion matrix.
17. Create a data frame containing two columns viz. name of the class and number of misclassifications.
18. Compare the performances of two different models that you have built and mention final conclusion.

Give your inferences at every step.