1. Predictive Maintenance of Power Substation Equipment by Infrared Thermography

This code is write for Predictive Maintenance of Power Substation Equipment by Infrared Thermography

Coding:

```
fs=fullfile('cnn','categories1');
catg={'Fault','Normal'};
imds=imageDatastore(fullfile(fs,catg),'IncludeSubfolders',1,'LabelSource','foldernames');
tbl=countEachLabel(imds);
minnum=min(tbl{:,2});
imds = splitEachLabel(imds,minnum,'randomize');
tbl=countEachLabel(imds);
net = resnet50();
[trainingset, testset]=splitEachLabel(imds,0.3,'randomize');
imagesize = net.Layers(1).InputSize;
augtrainset=augmentedImageDatastore(imagesize,trainingset,'colorPreprocessing','gray2rgb');
augtestset=augmentedImageDatastore(imagesize,testset,'colorPreprocessing','gray2rgb');
featureLayer='fc1000';
trainingFeatures=activations(net,augtrainset,featureLayer,'MiniBatchSize',32,'OutputAs','colu
mns');
trainLabels=trainingset.Labels;
classifier=fitcecoc(trainingFeatures,trainLabels,'Learner','Linear','Coding','onevsall','Observati
onsIn','columns');
testFeatures=activations(net,augtestset,featureLayer,'MiniBatchSize',32,'OutputAs','columns');
predictLabels=predict(classifier, testFeatures,'ObservationsIn','columns');
testLabels=testset.Labels;
confmat=confusionmat(testLabels,predictLabels);
confmat = bsxfun(@rdivide,confmat,sum(confmat,2));
mean(diag(confmat));
newim=imread(fullfile('test1.png'));
ds=augmentedImageDatastore(imagesize,newim,'colorPreprocessing','gray2rgb');
newfeature=activations(net,ds,featureLayer,'MiniBatchSize',32,'OutputAs','columns');
label = predict(classifier, newfeature,'ObservationsIn','columns');
sprintf('The given image is of Number %s ',label)
imshow('test1.png')
```

2. Renewable power generation via animate prime movers

This code is write for Developing Exercise and generation plan
To develope this Exercise/ Energy Generation Plan ,the physiological parameters such as fatigue level of the subject may need to be monitored; this parameter may be used as an indication of exhaustion which may be again subject to the weather condition, where an exhibition is reach early in hot/humid ambience and vice versa. So we have to design a mechanism of automation on the physiological parameters such as fatigue level and adjusting the power generation to animal physiological parameter.

2.1: Monitoring Voltage Level

We monitored the physiological parameters of the subject by monitoring the voltage level of the power generation cart. If the fatigue level of the animal is normal then the subject can exercise with his full strength and generate 12 volt dc by moving on circular path. As fatigue level increases, the ultimate power generation will decrease. So in keeping view this scenario , we develop a mechanism of automation that will run to physiological parameters to generate power by animal exercise as much as it can .

2.2: Voltage level and Microcontroller

Sensor will Measure voltage level and send data to microcontroller and Microcontroller decides which load to be cut off according to the data send by sensor to microcontroller on the basis of Adjus value of sensor as decides the power need to be increased or decreased. At normal fatigue level, the animal will move on the circular path with his full strength and generate 12 volt dc and this power supplied to all loads.

2.3: Movement/walking effect on animal

By moving on circular path, when fatigue level of the animal is increased and ultimately voltage level decreased wherein microcontroller decides that the voltage sensor values decreased from the adjusted normal value it will cut of 50% load from supplied power supply via relay. For further increase in fatigue level the voltage level also further decreased and the remaining 50% load will cut-off.

1. The power generation cart set-up of our project.
2. 12 V DC to 230V AC inverter setup
3. 50% our critical load
4.50% our non-critical load
5. Green led indicates normal fatigue level of the animal
6. Yellow led indicates average fatigue level of the animal
7. Red led indicates below average fatigue level of the animal
8. Speaker that is used to boast up the speed as our animal is voiced trained and boast up its speed on certain voice
9.&10 liquid crystal display with I2c module(LCD)to display the generated voltage
11.Microcontoller
12. voltage sensor

Coding:

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2); // I2C address 0x27, 16 column and 2 rows
long volt; // initialization
#define Switch1 2 // defining output pin for relay
```

```
#define Switch2 3 // defining output pin for relay
#define Switch3 4 // defining output pin for relay
#define Ledgreen 8 // defining output pin led
#define Ledyellow 9 // defining output pin led
#define Ledred 10 // defining output pin led
#define Ledblue 7 // defining output pin speaker MP3 module
#define Sensor A0 // defining input pin for sensing voltage
float vIN = 0; // initialization for input voltage
int a = 0; // initialization and declaration for check statment
void setup() {
lcd.init(); // initialize the lcd
lcd.backlight();
pinMode(Sensor,INPUT); // declaring input
pinMode(Switch1,OUTPUT); // declaring output
pinMode(Switch2,OUTPUT); // declaring output
pinMode(Switch3,OUTPUT); // declaring output
pinMode(Ledgreen,OUTPUT); // declaring output
pinMode(Ledyellow,OUTPUT); // declaring output
pinMode(Ledred,OUTPUT); // declaring output
pinMode(Ledblue,OUTPUT); // declaring output
a=0;
digitalWrite(Switch1,HIGH); // To write a HIGH or a LOW value to a digital pin.
digitalWrite(Switch2,HIGH); // To write a HIGH or a LOW value to a digital pin.
digitalWrite(Switch3,HIGH);// To write a HIGH or a LOW value to a digital pin.
}
void loop() {
int value = analogRead(Sensor); // read input voltage value
vIN = value * (5.0/1023.0) * 100.46;
lcd.setCursor(0, 0); // move cursor to (0, 0)
lcd.print("Voltage :"); // print message at (0, 0)
lcd.setCursor(10, 1); // move cursor to (2, 1)
lcd.print(vIN); // print message at (2, 1)
Serial.print("Voltage : ");
Serial.println(vIN);
delay(500);
if(a==0){
if(vIN>=12){ // if condition
digitalWrite(Ledgreen,HIGH);
digitalWrite(Ledyellow,LOW);
digitalWrite(Ledblue,LOW);
digitalWrite(Ledred,LOW);
digitalWrite(Switch1,HIGH);
digitalWrite(Switch2,HIGH);
digitalWrite(Switch3,HIGH);
}
if(vIN>=5 && vIN<=9){ // if condition
digitalWrite(Ledgreen,LOW);
digitalWrite(Ledyellow,HIGH);
digitalWrite(Ledblue,HIGH);
digitalWrite(Ledred,LOW);
```

```
digitalWrite(Switch1,HIGH);
digitalWrite(Switch2,LOW);
digitalWrite(Switch3,LOW);
delay(500);
digitalWrite(Ledblue,LOW);
delay(30000);
digitalWrite(Ledblue,HIGH);
delay(500);
digitalWrite(Ledblue,LOW);
}
if(vIN>2 && vIN<5){ // if condition
digitalWrite(Ledgreen,LOW);
digitalWrite(Ledyellow,LOW);
digitalWrite(Ledblue,LOW);
digitalWrite(Ledred,HIGH);
digitalWrite(Switch1,LOW);
digitalWrite(Switch2,LOW);
digitalWrite(Switch3,LOW);
a=1;
}}}
```