***PROBLEM STATEMENT*PROBLEM:**

Wastage of electrical energy.

**SOLUTION:**

To make a adaptive use of electrical energy. That is, when a person enters a room fans and lights can be set on automatically. When the person leaves the room, it can be set off. In case of huge auditorium, based on the persons in a particular area can be set on.

***IMPLEMENTATION IN AI:***

Human gesture can be fed as dataset and it can be implemented.

***CODE:***

|  |
| --- |
|  |
|  | import click |
|  | import os |
|  | import re |
|  | import face\_recognition.api as face\_recognition |
|  | import multiprocessing |
|  | import sys |
|  | import itertools |
|  |  |
|  |  |
|  | def print\_result(filename, location): |
|  | top, right, bottom, left = location |
|  | print("{},{},{},{},{}".format(filename, top, right, bottom, left)) |
|  |  |
|  |  |
|  | def test\_image(image\_to\_check, model): |
|  | unknown\_image = face\_recognition.load\_image\_file(image\_to\_check) |
|  | face\_locations = face\_recognition.face\_locations(unknown\_image, number\_of\_times\_to\_upsample=0, model=model) |
|  |  |
|  | for face\_location in face\_locations: |
|  | print\_result(image\_to\_check, face\_location) |
|  |  |
|  |  |
|  | def image\_files\_in\_folder(folder): |
|  | return [os.path.join(folder, f) for f in os.listdir(folder) if re.match(r'.\***\.**(jpg|jpeg|png)', f, flags=re.I)] |
|  |  |
|  |  |
|  | def process\_images\_in\_process\_pool(images\_to\_check, number\_of\_cpus, model): |
|  | if number\_of\_cpus == -1: |
|  | processes = None |
|  | else: |
|  | processes = number\_of\_cpus |
|  |  |
|  | # macOS will crash due to a bug in libdispatch if you don't use 'forkserver' |
|  | context = multiprocessing |
|  | if "forkserver" in multiprocessing.get\_all\_start\_methods(): |
|  | context = multiprocessing.get\_context("forkserver") |
|  |  |
|  | pool = context.Pool(processes=processes) |
|  |  |
|  | function\_parameters = zip( |
|  | images\_to\_check, |
|  | itertools.repeat(model), |
|  | ) |
|  |  |
|  | pool.starmap(test\_image, function\_parameters) |
|  |  |
|  |  |
|  | @click.command() |
|  | @click.argument('image\_to\_check') |
|  | @click.option('--cpus', default=1, help='number of CPU cores to use in parallel. -1 means "use all in system"') |
|  | @click.option('--model', default="hog", help='Which face detection model to use. Options are "hog" or "cnn".') |
|  | def main(image\_to\_check, cpus, model): |
|  | # Multi-core processing only supported on Python 3.4 or greater |
|  | if (sys.version\_info < (3, 4)) and cpus != 1: |
|  | click.echo("WARNING: Multi-processing support requires Python 3.4 or greater. Falling back to single-threaded processing!") |
|  | cpus = 1 |
|  |  |
|  | if os.path.isdir(image\_to\_check): |
|  | if cpus == 1: |
|  | [test\_image(image\_file, model) for image\_file in image\_files\_in\_folder(image\_to\_check)] |
|  | else: |
|  | process\_images\_in\_process\_pool(image\_files\_in\_folder(image\_to\_check), cpus, model) |
|  | else: |
|  | test\_image(image\_to\_check, model) |
|  |  |
|  |  |
|  | if \_\_name\_\_ == "\_\_main\_\_": |
|  | main() |