1. INTRODUCTION

1.1 Definition

What does Image Filter mean?

An image filter is a technique through which size, colors, shading and other characteristics of an image are altered. An image filter is used to transform the image using different graphical editing techniques. Image filters are usually done through graphic design and editing software.

Image filters are primarily used to edit an image using computer software. An image filter generally changes the image at the pixel level, meaning each pixel individually is affected. It can be applied to 2-D and 3-D images. Typically, the image filter process includes options such as:

Editing the color scheme/theme/contrast of the image

- Adjusting image brightness
- Adding effects to the image
- Changing the texture

The term image filter may also refer to the process of filtering (excluding) images from data, folders or Web searches.

1.2 Objective:

The use of image fiters now a days has become a very popular thing because of the social media apps that are being used by the present day world. Irrespective of their ages, everyone person wants to look cool in the social media. So, we are coming up with the idea of image filters which is trending in the present day world. A social media without its own image filters is absolutely worthless using it. Our objective is to come up with an interface where the user can edit pictures with filters provided by us.

1.3 Requirements for installing PyCharm:

Windows:

Microsoft Windows 10/8/7/Vista/2003/XP (incl.64-bit)

1 GB RAM minimum

2 GB RAM recommended

1024x768 minimum screen resolution

Python 2.4 or higher, Jython, PyPy or IronPython

Mac:

Mac OS X 10.8 or higher

1 GB RAM minimum

2 GB RAM recommended

Python 2.4 or higher, Jython, PyPy or IronPython

Linux:

512 MB RAM minimum, 1 GB RAM recommended

1024x768 minimum screen resolution

Python 2.4 or higher, Jython, PyPy or IronPython

2. ABOUT PYTHON

2.1 What is Python?

Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale.

Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library.

Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

Python is intended to be a highly readable language. It is designed to have an uncluttered visual layout, often using English keywords where other languages use punctuation. Python does not use curly brackets to delimit blocks, and semicolons after statements are optional, in contrast to many other programming languages. Further, Python has fewer syntactic exceptions and special cases than C or Pascal.

Python uses whitespace indentation to delimit blocks – rather than curly braces or keywords. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block.This feature is also sometimes termed the off-side rule.

Python's statements include (among others):

The assignment statement (token '=', the equals sign). This operates differently than in traditional imperative programming languages, and this fundamental mechanism (including the nature of Python's version of variables) illuminates many other features of the language. Assignment in C, e.g., x = 2, translates to "typed variable name x receives a copy of numeric value 2". The (right-hand) value is copied into an allocated storage location for which the (left-hand) variable name is the symbolic address. The memory allocated to the variable is large enough (potentially quite large) for the declared type. In the simplest case of Python assignment, using the same example, x = 2, translates to "(generic) name x receives a reference to a separate, dynamically allocated object of numeric (int) type of value 2." This is termed binding the name to the object. Since the name's storage location doesn't contain the indicated value, it is improper to call it a variable. Names may be subsequently rebound at any time to objects of greatly varying types, including strings, procedures, complex objects with data and methods, etc. Successive assignments of a common value to multiple names, e.g., x = 2; y = 2; z = 2 result in allocating storage to (at most) three names and one numeric object, to which all three names are bound. Since a name is a generic reference holder it is unreasonable to associate a fixed data type with it. However at a given time a name will be bound to some object, which will have a type; thus there is dynamic typing.

The if statement, which conditionally executes a block of code, along with else and elif (a contraction of else-if).

The for statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block.

The while statement, which executes a block of code as long as its condition is true.

The try statement, which allows exceptions raised in its attached code block to be caught and handled by except clauses; it also ensures that clean-up code in a finally block will always be run regardless of how the block exits.

The class statement, which executes a block of code and attaches its local namespace to a class, for use in object-oriented programming.

The def statement , which defines a function or method.

The with statement , which encloses a code block within a context manager (for example, acquiring a lock before the block of code is run and releasing the lock afterwards, or opening a file and then closing it), allowing Resource Acquisition Is Initialization (RAII)-like behavior.

The pass statement, which serves as a NOP. It is syntactically needed to create an empty code block.

The assert statement, used during debugging to check for conditions that ought to apply.

The yield statement, which returns a value from a generator function. From Python 2.5, yield is also an operator. This form is used to implement coroutines.

The import statement, which is used to import modules whose functions or variables can be used in the current program.There are two ways of using import . from <module name> import * or import <module name>.

The print statement was changed to the print() function in Python 3.

Python does not support tail call optimization or first-class continuations, and, according to Guido van Rossum, it never will.However, better support for coroutine-like functionality is provided in 2.5, by extending Python's generators. Before 2.5, generators were lazy iterators; information was passed unidirectionally out of the generator. From Python 2.5, it is possible to pass information back into a generator function, and from Python 3.3, the information can be passed through multiple stack levels.

Expressions:

Some Python expressions are similar to languages such as C and Java, while some are not:

Addition, subtraction, and multiplication are the same, but the behavior of division differs. Python also added the ** operator for exponentiation.

From Python 3.5, it enables support of matrix multiplication with the @ operator.

In Python, == compares by value, versus Java, which compares numerics by value and objects by reference. (Value comparisons in Java on objects can be performed with the equals() method.) Python's is operator may be used to compare object identities (comparison by reference). In Python, comparisons may be chained, for example a <= b <= c.

Python uses the words and, or, not for its boolean operators rather than the symbolic &&, ||, ! used in Java and C.

Python has a type of expression termed a list comprehension. Python 2.4 extended list comprehensions into a more general expression termed a generator expression.Anonymous functions are implemented using lambda expressions; however, these are limited in that the body can only be one expression.

Conditional expressions in Python are written as x if c else y (different in order of operands from the c ? x : y operator common to many other languages).

Python makes a distinction between lists and tuples. Lists are written as [1, 2, 3], are mutable, and cannot be used as the keys of dictionaries (dictionary keys must be immutable in Python). Tuples are written as (1, 2, 3), are immutable and thus can be used as the keys of dictionaries, provided all elements of the tuple are immutable. The + operator can be used to concatenate two tuples, which does not directly modify their contents, but rather produces a new tuple containing the elements of both provided tuples. Thus, given the variable t initially equal to (1, 2, 3), executing t = t + (4, 5) first evaluates t + (4, 5), which yields (1, 2, 3, 4, 5), which is then assigned back to t, thereby effectively "modifying the contents" of t, while conforming to the immutable nature of tuple objects. Parentheses are optional for tuples in unambiguous contexts.

Python features sequence unpacking where multiple expressions, each evaluating to anything that can be assigned to (a variable, a writable property, etc), are associated in the identical manner to that forming tuple literals and, as a whole, are put on the left hand side of the equal sign in an assignment statement. The statement expects an iterable object on the right hand side of the equal sign that produces the same number of values as the provided writable expressions when iterated through, and will iterate through it, assigning each of the produced values to the corresponding expression on the left.

Python has a "string format" operator %. This functions analogous to printf format strings in C, e.g. "spam=%s eggs=%d" % ("blah", 2) evaluates to "spam=blah eggs=2". In Python 3 and 2.6+, this was supplemented by the format() method of the str class, e.g. "spam={0} eggs={1}".format("blah", 2), Python 3.6 added "f-strings": f'spam={"blah"} eggs={2}'.

Python has various kinds of string literals:

Strings delimited by single or double quote marks. Unlike in Unix shells, Perl and Perl-influenced languages, single quote marks and double quote marks function identically. Both kinds of string use the backslash (\) as an escape character. String interpolation became available in Python 3.6 as "formatted string literals".

Triple-quoted strings, which begin and end with a series of three single or double quote marks. They may span multiple lines and function like here documents in shells, Perl and Ruby.

Raw string varieties, denoted by prefixing the string literal with an r. Escape sequences are not interpreted; hence raw strings are useful where literal backslashes are common, such as regular expressions and Windows-style paths. Compare "@-quoting" in C#.

Python has array index and array slicing expressions on lists, denoted as a[key], a[start:stop] or a[start:stop:step]. Indexes are zero-based, and negative indexes are relative to the end. Slices take elements from the start index up to, but not including, the stop index. The third slice parameter, called step or stride, allows elements to be skipped and reversed. Slice indexes may be omitted, for example a[:] returns a copy of the entire list. Each element of a slice is a shallow copy.

In Python, a distinction between expressions and statements is rigidly enforced, in contrast to languages such as Common Lisp, Scheme, or Ruby. This leads to duplicating some functionality. For example:

List comprehensions vs. for-loops

Conditional expressions vs. if blocks

The eval() vs. exec() built-in functions (in Python 2, exec is a statement); the former is for expressions, the latter is for statements.

Statements cannot be a part of an expression, so list and other comprehensions or lambda expressions, all being expressions, cannot contain statements. A particular case of this is that an assignment statement such as a = 1 cannot form part of the conditional expression of a conditional statement. This has the advantage of avoiding a classic C error of mistaking an assignment operator = for an equality operator == in conditions: if (c = 1) { ... } is syntactically valid (but probably unintended) C code but if c = 1: ... causes a syntax error in Python.

Methods

Methods on objects are functions attached to the object's class; the syntax instance.method(argument) is, for normal methods and functions, syntactic sugar for Class.method(instance, argument). Python methods have an explicit self parameter to access instance data, in contrast to the implicit self (or this) in some other object-oriented programming languages (e.g., C++, Java, Objective-C, or Ruby).

2.2 Libraries

Python has a large standard library, commonly cited as one of Python's greatest strengths, providing tools suited to many tasks. This is deliberate and has been described as a "batteries included" Python philosophy. For Internet-facing applications, many standard formats and protocols (such as MIME and HTTP) are supported. Modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals.

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation wsgiref follows PEP 333), but most modules are not.

3. About PIL

3.1 Introduction

Python Imaging Library (abbreviated as PIL) (in newer versions known as Pillow) is a free library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats. It is available for Windows, Mac OS X and Linux. The latest version of PIL is 1.1.7, was released in September 2009 and supports Python 1.5.2–2.7, with Python 3 support to be released "later".

3.2 Capabilities

Pillow offers several standard procedures for image manipulation. These include:

- per-pixel manipulations,
- masking and transparency handling,
- image filtering, such as blurring, contouring, smoothing, or edge finding
- image enhancing, such as sharpening, adjusting brightness, contrast or color
- adding text to images and much more.

3.3 File formats

Some of the file formats supported include PPM, PNG, JPEG, GIF, TIFF, and BMP. It is also possible to create new file decoders to expand the library of file formats accessible.

3.4 Usage example

```
from PIL import Image, ImageFilter  # imports the library


original = Image.open("file.ppm") # load an image from the hard drive

blurred = original.filter(ImageFilter.BLUR) # blur the image

original.show() # display both images

blurred.show()
```

4. About Tkinter

4.1 Introduction

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. Tkinter is included with the standard Microsoft Windows and Mac OS X install of Python.

As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.

4.2 Window

This term has different meanings in different contexts, but in general it refers to a rectangular area somewhere on the user's display screen.

4.3 Top Level Window

A window that exists independently on the screen. It will be decorated with the standard frame and controls for the desktop manager. It can be moved around the desktop, and can usually be resized.

4.4 Widget

The generic term for any of the building blocks that make up an application in a graphical user interface. Examples of widgets: buttons, radiobuttons, text fields, frames, and text labels.

4.5 Frame

In Tkinter, the Frame widget is the basic unit of organization for complex layouts. A frame is a rectangular area that can contain other widgets.

4.6 Child and parent

When any widget is created, a parent-child relationship is created. For example, if you place a text label inside a frame, the frame is the parent of the label.

A minimal application

```
 1 #!/usr/bin/env python3
 2 import tkinter as tk
 3
 4 class Application(tk.Frame):
 5     def __init__(self, master=None):
 6         tk.Frame.__init__(self, master)
 7         self.grid()
 8         self.createWidgets()
 9
10     def createWidgets(self):
11         self.quitButton = tk.Button(self, text='Quit', command=self.quit)
12         self.quitButton.grid()
13
14 app = Application()
15 app.master.title('Sample application')
16 app.mainloop()
```

5. IMPLEMENTATION

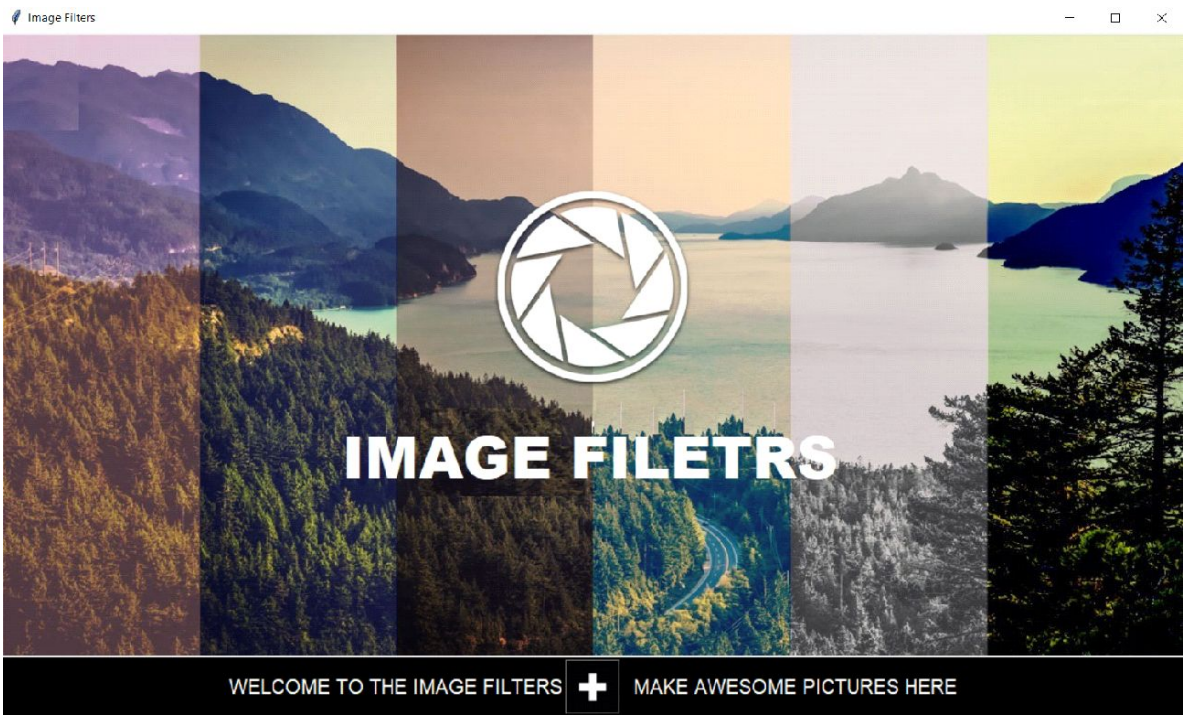
5.1 Introduction

The project when run it first asks the user to select a picture. When picture is selected along with the picture selected two options namely FILTERS and CROP appears on the screen. As per the user's choice when filters are selected project provides various filters.User has a feasibility to save picture according to the filter's enhancement or one can go back and check other filters as there is always an option to go back or one can select crop option which has many croping options provided for croping pictures efficiently.

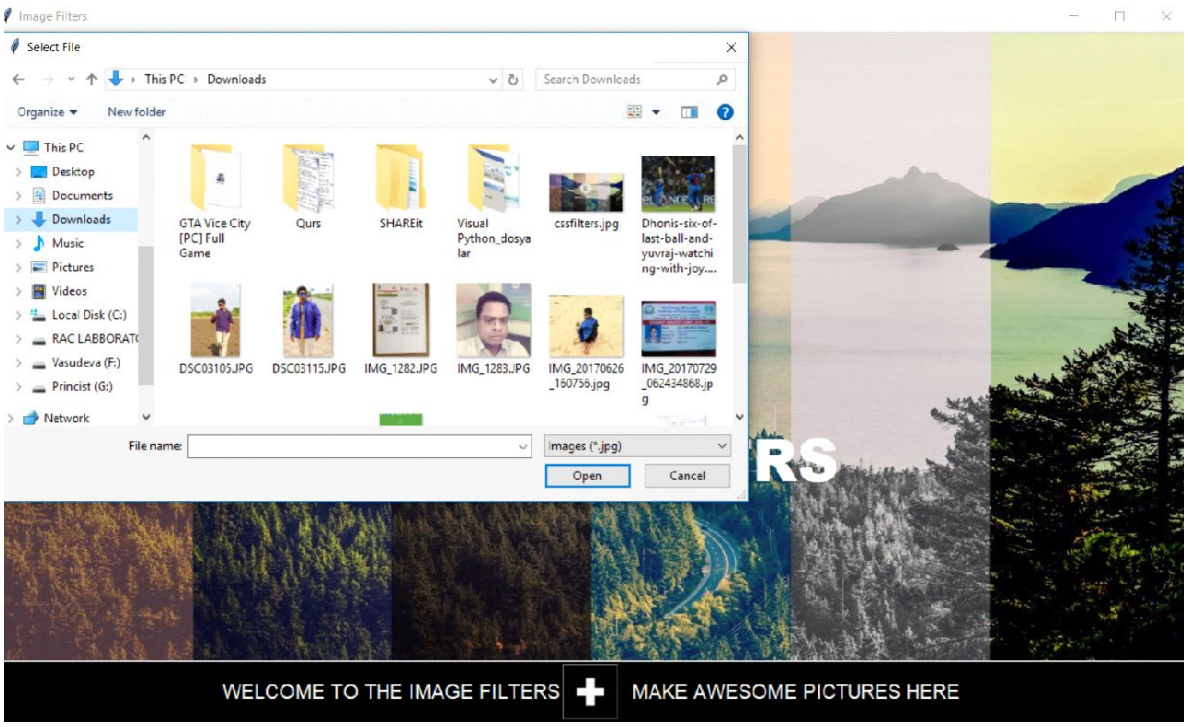
5.2 The System

The following information will show the working of ‘Image Filters ’ along with relevant screen shots.

5.2.1 Main screen



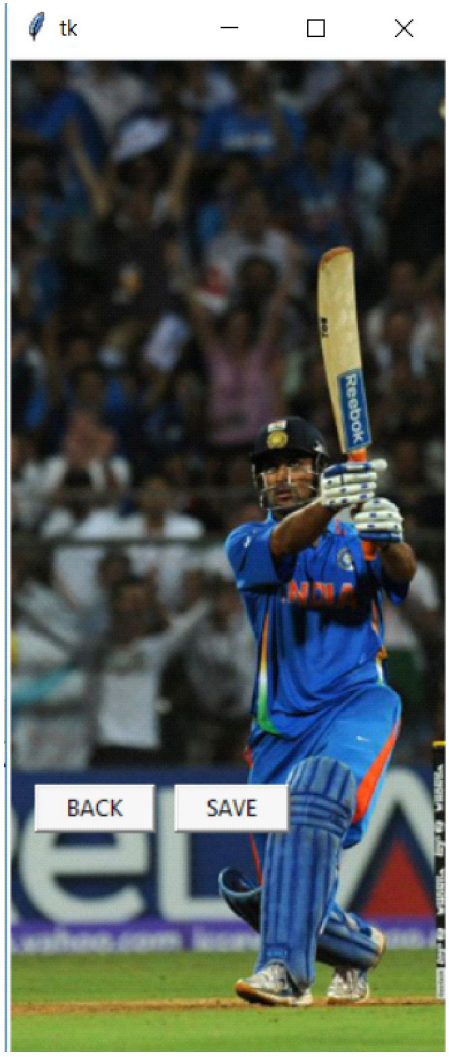
5.2.2 Selecting Picture



5.2.3 Option for selection filters or crop for selected picture

5.2.4 Few filters

5.2.5 Few crops


tk
Reebok
BACK
SAVE


tk
YUVRAJ
12
BACK
SAVE

6. Conclusion and future scope

Although project "Image filters" prodvide various types of filters for image enhancement and croping there is always a scope for improvement as project as depth. As PIL is a huge topic to dive in one can obtain uncoutable image enchancing procedures and filters. So the future scope of project is rich. By doing this kind of projects one could get good knowledge about Python and it's GUI as it has become programming world's favourite.

7.Bibliography

1. https://www.wikipedia.org/
2. https://www.python.org/
3. http://pillow.readthedocs.io/en/3.4.x/reference/ImageFilter.html
4. 3http://www.w3schools.com/