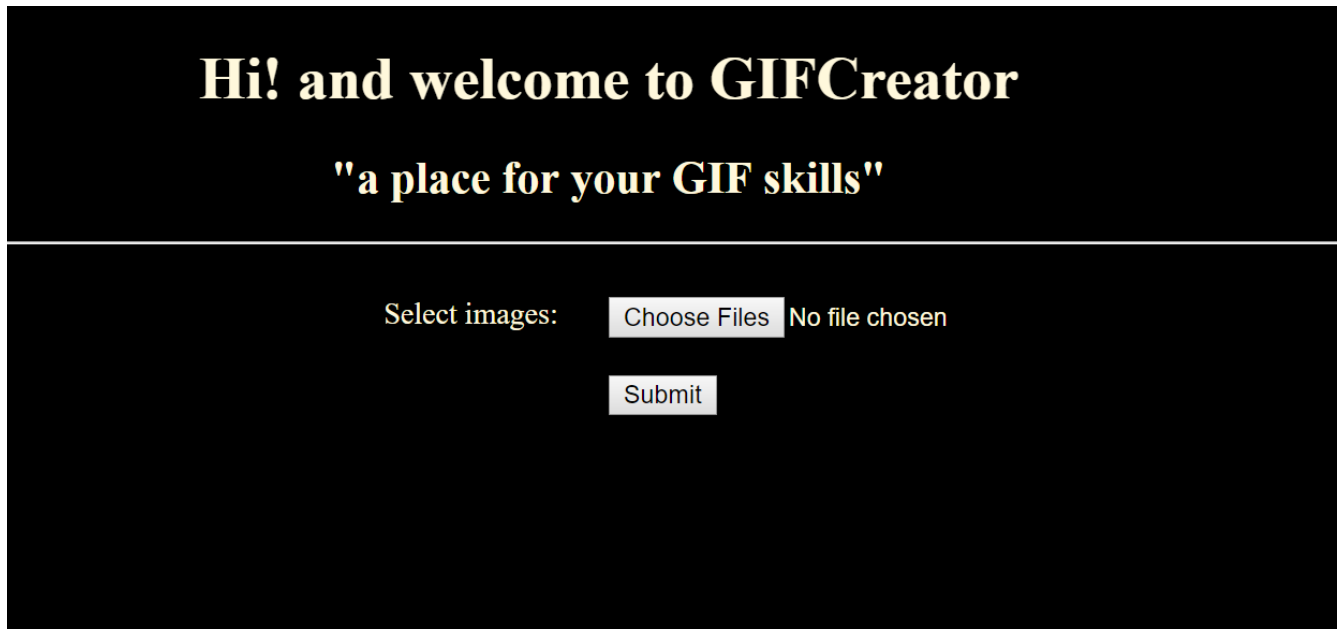


Referat

Crearea unui GIF



GIFCreator se numeste aplicatia si are ca scop crearea unui GIF printr-un upload de imagini. Aplicatia are la baza o interfata web, un server API scris in python pe frameworkul Flask dar si alte librarii necesare precum ImageIO, numpy si PIL.

Utilizatorul la deschidere paginii home are la dispozitie doua butoane, unul prin care poate incarca mai multe fisiere odata si un buton de submit care trimite un request spre API pentru a genera giful dorit.

Dupa ce giful a fost creat, utilizatorul este trimis pe o noua pagina unde poate vizualiza giful creat, in acelasi timp il poate si descarca.

Here is your result

"a great one by the way"



```
def prepare_image(image, target):
    # if img is not RGB, convert it
    if image.mode != "RGB":
        image = image.convert("RGB")

    image = image.resize(target)
    # image = imagenet_utils.preprocess_input(image)

    return image

@app.route("/load_images", methods=['POST'])
def load_images():
    images = []
    files = request.files.getlist("images[]")
    for file in files:
        image = Image.open(io.BytesIO(file.read()))
        image = prepare_image(image, (299, 299))
        image = np.asarray(image)
        images.append(image)

    imageio.mimsave('static/gif_to_display.gif', images, format='GIF', duration=1)

    return render_template('Result.html')
```

Imageio are un nucleu relativ simplu care oferă o interfață comună la diferite formate de fișiere. Acest nucleu se ocupă de citirea din diferite surse (cum ar fi http) și expune un simplu API pentru ca pluginurile să poată accesa datele brute. Toate formatele de fișiere sunt implementate în pluginuri. Plugurile suplimentare pot fi înregistrate cu ușurință.

Unele pluginuri se bazează pe biblioteci externe (de ex. Ffmpeg). Imageio oferă o modalitate de descărcare a acestora cu un apel de o funcție și îi solicită utilizatorului să facă acest lucru atunci când este necesar. Descărcarea este stocată în cache în directorul appdata, acest lucru păstrează imaginea luminoasă și scalabilă.

Imageio oferă o gamă largă de formate de imagine, inclusiv formate științifice. Orice ajutor cu implementarea mai multor formate este foarte binevenit!

Baza de coduri aderă la (un subset de) ghidajele stilului PEP8. Ne străduim pentru acoperirea maximă a testelor (100% pentru nucleu, > 95% pentru fiecare plugin).

Imageio sa bazat pe frustrarea pe care multe biblioteci care aveau nevoie să le citească sau să scrie date de imagine și-au produs propriile funcționalități pentru IO. PIL nu a satisfăcut foarte bine nevoile, iar bibliotecile precum scikit-image trebuie să poată face față formatelor științifice. A fost nevoie de o imagine bună în bibliotecă, care este o dependență ușoară, ușor de întreținut și scalabilă în formate de fișiere exotice.

Imageio a început ca o componentă a proiectului scikit-image, prin care a fost capabil să suporte o mulțime de formate comune. Am creat un nucleu simplu, dar puternic, un utilizator API curat și un sistem adecvat de pluginuri.

Scopul imaginii este de a sprijini citirea și scrierea imaginilor. Nu procesăm imagini, trebuie să utilizați ex. imaginație pentru asta. Imageio trebuie să fie ușor de instalat și să fie ușor. Sistemul de plugin Imageio permite să scalați numărul de formate acceptate și să păstrați încă o amprentă redusă. Este speranța noastră de a forma un grup de dezvoltatori, fiecare dintre care să mențină unul sau mai multe pluginuri. În acest fel, burgerul fiecărui dezvoltator este scăzut și, împreună, putem face imaginea într-o bibliotecă