COLEGIUL NATIONAL „ELENA CUZA”  
**BUCURESTI**

**Sintetica**

Elev: Profesor coordonator:

**Sturza Mihai-George Mirzacu Marius-Emilian**

# Cuprins

1. Abstract
2. Generalități despre tehnologiile folosite
   1. JavaScript
      1. NodeJS
      2. ReactJS
      3. Bootstrap
   2. Python
3. Flask
4. Tensorflow
5. PyTorch
6. TextBlob
   1. Git
7. Generalități despre algoritmii folosiți
8. StyleTransfer
9. Progressive Growing of GANs
10. GPT-2
11. Modul de operare
12. Modulul „frontend”
13. Componentele website-ului
14. Modul de interacționare cu modulul „backend”
15. Modulul „backend”
16. Inițializarea sistemelor
17. Modul de furnizare al datelor (API)
18. Testarea sistemelor
19. Vizual
20. Postman

# 1. Abstract

Sintetica este o interfață web care comunică cu algoritmi puternici și moderni din domeniul inteligenței artificiale. Scopul proiectului este de a informa despre ce pot cu adevărat astfel de sisteme inteligente într-un mod cat mai simplu pentru utilizator.

Sintetica este un semnal de alarmă pentru riscul ce îl prezinta folosirea acestor algoritmi de persoane nepotrivite, care au alte scopuri decât cele educaționale. Putem vedea o mulțime de astfel de pericole în jurul nostru: de la Deep Fakes la Fake News, care sunt ajutate de astfel de algoritmi generativi.

# 2. Generalități despre tehnologiile folosite

# 2.1 Javascript

2.1.1 NodeJS este un runtime JavaScript asincron condus de evenimente, conceput pentru a crea aplicații de rețea scalabile. Acest lucru este în contrast cu modelul de concurență mai obișnuit de astăzi, în care sunt utilizate „threads”-uri de SO. Rețeaua bazată pe „threads” este relativ ineficientă și foarte dificil de utilizat. În plus, utilizatorii Node.js sunt liberi de îngrijorările legate de blocarea procesului, deoarece nu există blocări. Aproape nici o funcție din Node.js nu efectuează direct I / O, deci procesul nu se blochează niciodată. [1] In acest proiect, NodeJS este baza pe care rulează ReactJS, componentă care desenează interfața website-ului.

2.1.2 React (cunoscut și sub numele de React.js sau ReactJS) este o bibliotecă JavaScript open-source, front-end, pentru a construi interfețe utilizator sau componente UI. Este întreținut de Facebook și de o comunitate de dezvoltatori și companii individuale. React poate fi folosit ca bază în dezvoltarea de aplicații cu o singură pagină sau mobile. [2] Pentru Sintetica, în React a fost făcută toata interfața vizuală cât si „Routing”-ul între pagini.

2.1.3 Bootstrap este un cadru CSS gratuit și open-source orientat către dezvoltarea web front-end receptivă, „mobile-first”. Conține șabloane de proiectare bazate pe CSS și (opțional) JavaScript pentru tipografie, formulare, butoane, navigare și alte componente ale interfeței. [3]

# 2.2 Python

2.2.1 Flask este un micro framework web scris în Python. Este clasificat ca micro framework, deoarece nu necesită anumite instrumente sau biblioteci. Nu are un strat de abstractizare a bazei de date, validarea formularului sau orice alte componente în care bibliotecile terțe preexistente oferă funcții comune. [4] În Sintetica, Flask se ocupă de toate cererile externe către modelele de inteligență artificială.

2.2.2 Tensorflow este o bibliotecă software gratuită și open-source pentru „machine learning”. Poate fi folosit într-o serie de sarcini, dar se concentrează în mod special pe „learning” și „inferece” a „deep neural networks”. [5] Modelul de StyleTransfer este făcut în această bibliotecă.

2.2.3 PyTorch este o bibliotecă open source de „machine learning” bazată pe biblioteca Torch, folosită pentru aplicații precum „computer vision” și procesarea limbajului natural, dezvoltată în principal de laboratorul de cercetare AI (FAIR) de la Facebook. [6] În Sintetica, PyTorch joacă un rol important pentru algoritmul de generare al imaginilor sintetice dar și pentru generarea textului.

2.2.4 TextBlob este o bibliotecă Python pentru prelucrarea datelor textuale. Oferă un API simplu pentru sarcini comune de procesare a limbajului natural (NLP), cum ar fi etichetarea parțială a vorbirii, extragerea frazelor substantivale, analiza sentimentelor, clasificarea, traducerea și multe altele. [7] Sintetica prezintă implementarea TextBlob în demo-ul „bonus”, pentru analiză textului.

2.3 Git

Git este un software pentru urmărirea modificărilor din orice set de fișiere, utilizat de obicei pentru coordonarea muncii între programatori care dezvoltă în colaborare codul sursă în timpul dezvoltării software-ului. Obiectivele sale includ viteza, integritatea datelor și asistența pentru fluxuri de lucru neliniare distribuite (mii de ramuri paralele care rulează pe sisteme diferite). [8] Tot proiectul a fost dezvoltat cu ajutorul acestui sistem, care a fost ținut de cei de la GitHub [9].

# 3. Generalități despre algoritmii folosiți

# 3.1 StyleTransfer

Neural Style Transfer (NST) se referă la o clasă de algoritmi software care manipulează imagini digitale sau videoclipuri, pentru a adopta aspectul sau stilul vizual al altei imagini. Algoritmii NST se caracterizează prin utilizarea rețelelor neuronale profunde de dragul transformării imaginii. Utilizările obișnuite pentru NST sunt crearea de lucrări de artă artificială din fotografii, de exemplu prin transferarea aspectului unor picturi celebre pe fotografii furnizate de utilizator. [10] NST se bazează pe algoritmi de sinteză a texturii bazate pe histograme, în special metoda lui Portilla și Simoncelli. NST poate fi rezumat ca sinteză de texturi bazată pe histograme cu caracteristici ale rețelei neuronale convoluționale (CNN) pentru problema analogiilor de imagine. Hârtia originală folosea o arhitectură VGG-19 [11] care a fost pregătită în prealabil pentru a efectua recunoașterea obiectelor folosind setul de date ImageNet.

# 3.2 Progressive Growing of GANs

Generative Adversarial Network (GAN) este o clasă de cadre de învățare automată proiectate de Ian Goodfellow și colegii săi în 2014. Două rețele neuronale se dispută între ele într-un joc (sub forma unui joc cu sumă zero, unde câștigul unui agent este pierderea unui alt agent). [12] Iar, ideea cheie pentru Progressive Growing of GANs este să creștem atât generatorul, cât și discriminatorul progresiv: pornind de la o rezoluție scăzută, adăugăm straturi noi care modelează detalii din ce în ce mai fine pe măsură ce antrenamentul progresează. Acest lucru accelerează atât antrenamentul, cât și îl stabilizează foarte mult, permițându-ne să producem imagini de calitate fără precedent, de exemplu, imagini CELEBA[13]. [14]

# 3.3 GPT-2

GPT-2 este un model de limbaj bazat pe „large transformer”, cu 1,5 miliarde de parametri, instruit pe un set de date de 8 milioane de pagini web. GPT-2 este instruit cu un obiectiv simplu: prezice cuvântul următor, având în vedere toate cuvintele anterioare din unele texte. Diversitatea setului de date face ca acest obiectiv simplu să conțină demonstrații naturale ale multor sarcini în diverse domenii. GPT-2 este o extindere directă a GPT, cu mai mult de 10X parametrii și instruită pe mai mult de 10X cantitatea de date. GPT-2 afișează un set larg de capabilități, inclusiv abilitatea de a genera eșantioane de text sintetic condiționate de calitate fără precedent, în cazul în care primăm modelul cu o intrare și îi facem să genereze o continuare lungă. În plus, GPT-2 depășește alte modele de limbă instruite pe domenii specifice (cum ar fi Wikipedia, știri sau cărți) fără a fi nevoie să utilizeze aceste seturi de date de formare specifice domeniului. [15]

# 4. Modul de operare

# 4.1 Modulul „frontend”

4.1.1 Componentele website-ului. Website-ul este format din 3 pagini.  
Prima pagina (atestat.mihaisturza.ro) constă în prezentarea informațiilor despre proiect. Aceasta conține un element „hero” care este văzut încă din primele momente pe pagina. Sub acesta, este plasat un element descriptiv despre autorul proiectului: Mihai-George Sturza. După, sunt prezentați algoritmii pe care ii putem testa in cadrul website-ului. In finalul paginii, putem observa un Call To Action(CTA) pentru descoperirea codului sursa, ce este ținut pe pagina de GitHub[9].  
A doua pagina (atestat.mihaisturza.ro/incearca) consta in înșiruirea „demo”-urilor disponibile prin poze, icons și animații.

A treia pagina este o pagina șablon, adaptata in funcție de „demo”-urile disponibile. Aceasta se poate găsi in 4 variații:

1. Generare de imagini(profil)  
   Pagina prezinta o scurta descriere a modului de funcționare alături de instrucțiuni pentru a obține rezultate de la sistemul inteligent. Pe pagina se afla un singur buton, pe care scrie „Generează”, care, odată apăsat, generează o poză de profil sintetică.
2. Generare de text  
   Pagina prezinta o scurta descriere a modului de funcționare alături de instrucțiuni pentru a obține rezultate de la sistemul inteligent. Pe pagină se afla o zonă de text, care cere un prefixul ce va fi furnizat către sistemul generator. Apăsarea pe butonul „Generează” va afișa datele create de algoritm.
3. Transfer de stil  
   Pagina prezinta o scurta descriere a modului de funcționare alături de instrucțiuni pentru a obține rezultate de la sistemul inteligent. Pe pagină se află două zone pentru încărcarea de fișiere, o imagine contextuala și una stilistică. Apăsarea pe butonul „Trimite” va combina stilul imaginii stilistice cu cea contextuală, afișând rezultatul final.
4. Analiză de sentiment și percepția brandului  
   Pagina prezinta o scurta descriere a modului de funcționare alături de instrucțiuni pentru a obține rezultate de la sistemul inteligent. Pe pagină se afla o zonă de text, care necesită o părere din partea unui utilizator asupra unui brand/produs/serviciu etc. existent sau inexistent. Trimiterea părerii nu va stoca nicio informație pe server și doar va întoarce analiza de sentiment, subiectivitate și semantică a propoziției date.

4.1.2 Modul de interacționare cu modului „backend”. Din punct de vedere tehnic, modulul „frontend” rulează pe un server propriu, de NodeJS, având grijă de propriile conexiuni și rute.

În modul de dezvoltare, acesta asculta pe portul 3000, iar in modul de producție, pe 80 sau 443, in funcție de protocolul disponibil (HTTP/HTTPS). Legătura dintre cele două module se face printr-un API pus la dispoziție de modului „backend”. Acesta se folosește ca un API web obișnuit, prin apelarea unor rute bine definite.

# 4.2 Modulul „Backend”

4.2.1 Inițializarea sistemelor se face prin selectare unui mod de rulare. Modulul este prevăzut cu 2 moduri predefinite: „Development” și „Production”. Ambele ascultă pe portul 5000 al calculatorului gazdă, fiind nevoie de un proxy sau un forward către porturile standard de HTTP sau HTTPS, în cazul în care nu este furnizat alt port prin variabila de mediu.  
Pe lângă partea de networking, modulul trebuie să încarce 3 modele statistice. Acestea vor fi descărcate la momentul pornirii, doar dacă este prima dată când modulul rulează. Dacă nu, acestea vor fi preluate din cel mai adecvat „cache”. Folosindu-se de librării precum PyTorch și Tensorflow, acesta servește modelele către clienți printr-un API.

4.2.2 Modul de furnizare al datelor se face printr-un API. Acesta are urmatoarea structură:

* /api (Blueprint)  
  Ruta-părinte pentru toate conexiunile referitoare la API.
  + /perception (POST)  
    Ruta pentru „demo”-ul „Analiză de sentiment și percepția brandului”
  + /textgen (POST)  
    Ruta pentru „demo”-ul „Generare de text”
  + /pgg (GET)  
    Ruta pentru „demo”-ul „Generare de imagini (profil)”
  + /transfer (POST)  
    Ruta pentru „demo”-ul „Transfer de stil”

Blueprint – crearea unei „harți” pentru rutele serverului, POST – cerere ce conține un corp de date și așteaptă un răspuns, GET – cerere ce nu trimite un corp de date, dar așteaptă răspuns.

Pentru a asigura funcționarea corectă a API-ului, toate răspunsurile date de către modul, sunt in format JSON[16], iar toate datele pe care le primește, sunt de așteptat să fie in același format. De asemenea, toate rutele respectă CORS (Cross-origin resource sharing) [17]

Transportul datelor de tip imagine se face prin structuri ce conțin date în „base64”, fiind reconstruite in date statistice.

# 4.3 Testarea sistemelor

4.2.1 Se poate face vizual, prin interfața web pusa la dispoziție. Rutele disponibile sunt:

* atestat.mihaisturza.ro
  + /
  + /incearca
    - /perceptie
    - /generator
    - /gans
    - /transfer

4.2.1 Se poate face programatic, prin orice tip de software care implementează API-ul descris sau prin Postman, un software dedicat testării API-urilor. Rutele disponibile sunt:

* /api (Blueprint)  
  Ruta-părinte pentru toate conexiunile referitoare la API.
  + /perception (POST)  
    Așteaptă un JSON ce conține „key”-ul „blob” cu informații de tip text. Dă inapoi un JSON ce conține „key”-urile: „sentences”, „tags, „polarity”, „subjectivity”, „summary”, „warnings”.
  + /textgen (POST)  
    Așteaptă un JSON ce conține „key”-ul „context” cu informații de tip text. Dă inapoi un JSON ce conține „key”-urile: „original”, „translated”, „errors”
  + /pgg (GET)  
    Trimite un JSON o imagine sub forma unui „blob” in base64 in „key”-ul „image”.
  + /transfer (POST)  
    Așteaptă un JSON ce conține „key”-urile „content”, „style” cu informații „blob” in base64. Dă inapoi un JSON ce conține o imagine sub forma unui „blob” in base64 in „key”-ul „image”.

Bibliografie

1. Despre NodeJS (<https://nodejs.org/en/about/>)
2. React (JavaScript Library) (<https://en.wikipedia.org/wiki/React_(JavaScript_library)>)
3. Bootstrap (front-end framework) (<https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)>)
4. Flask (web framework) (<https://en.wikipedia.org/wiki/Flask_(web_framework)>)
5. TensorFlow (<https://en.wikipedia.org/wiki/TensorFlow>)
6. PyTorch (<https://en.wikipedia.org/wiki/PyTorch>)
7. TextBlob: Simplified Text Processing (<https://textblob.readthedocs.io/en/dev/>)
8. Git (<https://en.wikipedia.org/wiki/Git>)
9. Sintetica on GitHub (<https://github.com/sturzamihai/sintetica>)
10. Neural Style Transfer (<https://en.wikipedia.org/wiki/Neural_Style_Transfer>)
11. VGG-19 (<https://www.robots.ox.ac.uk/~vgg/research/very_deep/>)
12. Generative adversarial network (<https://en.wikipedia.org/wiki/Generative_adversarial_network>)
13. CelebA Dataset (<http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>)
14. Progressive Growing of GANs (<https://arxiv.org/pdf/1710.10196.pdf>)
15. Better language models (<https://openai.com/blog/better-language-models/>)
16. JSON (<https://www.json.org/json-en.html>)
17. Cross-origin resource sharing (<https://en.wikipedia.org/wiki/Cross-origin_resource_sharing>)