

# AM Tema 2

- Drăgan Pavel 331AA -

## Generarea imaginii

Pentru generarea imaginii în funcție de detaliile asignate pentru tema am folosit modelul de text-to-image Stable Diffusion recomandat în cerința temei, implementat la fel ca în resursa listată. O observație personală pe care o pot face legată de model este că în cazul în care nu era menționat un “art style” specific, cel puțin în cazul prompt-urilor mele, acesta tindea spre un stil mai “cartoonish”. De asemenea a trebuit să găsesc un echilibru între un prompt sumar și unul foarte detaliat deoarece pare că nici o extremă nu produce rezultatele ideale, cele mai bune imagini fiind acelea cu un număr moderat de detalii în prompt. (Am generat imaginea folosind alt fișier. Am adăugat codul ca să respect cerințele de upload. Se poate comenta partea de generare de la început pentru verificarea celorlalte funcții ale programului).



Figure 1 Imagine fara "art style" detaliat



Figure 2 Imagine cu "art style" realistic și un număr echilibrat de detalii în prompt

## Spatiul de culoare

Spatiul de culoare asignat pentru mine a fost YUV. Acesta era spatiul de culoare folosit în principal pentru televizoarele analog deoarece putea decoda și afișa atât imagini alb negru cât și color.

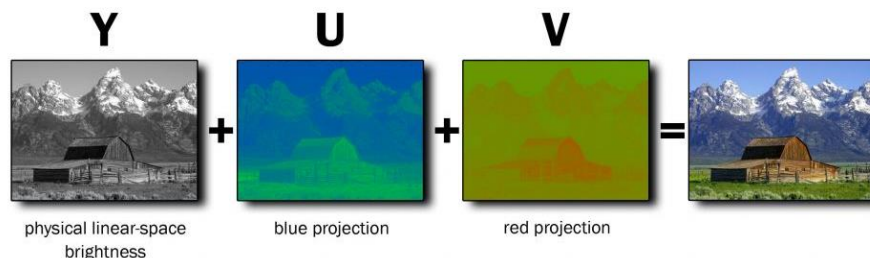


Figure 3 Reprezentare a celor 3 canale

O culoare este descrisă ca o componentă  $Y'$  (luma) și două componente cromatice  $U$  și  $V$ . Simbolul de derivare ( $'$ ) indică faptul că  $Y$  este calculat dintr-un input “gamma-corrected” RGB și că este diferită de luminanța (luminance) reală.

După ce am făcut conversia folosind funcția din cv2 `COLORBGR2YUV` (am făcut din bgr deoarece cv2 face conversia automat din rgb în bgr atunci când citește imagini) și a trebuit să salvez imaginile cu spațiul de culoare YUV am decis să le salvez folosind extensia .tiff deoarece este un format cu compresie lossless care este cunoscut pentru calitatea imaginilor și potrivit în cazul în

care utilizatorul vrea sa mai editeze pozele respective. Desi si png este lossless este asociat mai mult cu spatial de culoare RGB asa ca nu am optat pentru el.

## **Detectarea obiectului**

Pentru detectarea obiectului am folosit MediaPipe Object Detector cu un model pre-antrenat de detectie a obiectelor care continea si o clasa cu placi de surf. Am editat parametrii ca acesta sa listeze doar un singur obiect si am scazut limita de siguranta in detectie deoarece placa de surf generata nu arata chiar identic cu una reala. De asemenea, deoarece a fost mentionat in cerinta sa respectam exact formatele si documentele cerute nu am urcat si fisierul .tflite de care acesta ar avea nevoie ca sa functioneze asa cum trebuie.

## **Extragerea obiectului si masca binara**

Pentru crearea mastii binare si extragerea obiectului am folosit doua metode. Prima metoda, asa cum e sugerat in laborator este cu o masca de culoare prin care dam un range pentru culoare (in spatiul hsv) pe care vrem sa o detectam si extragem datele necesare. Nu am fost complet multumit de aceasta metoda asa ca am incercat si o metoda folosind functia de grab cut din OpenCV aplicata pentru bounding box-ul stabilit in etapa de detectare a obiectului. In acest moment este folosita a doua metoda dar si prima este prezenta in cod si doar trebuie comentata si apoi commentate liniile ce tin de a doua metoda.

## **Sintetizatorul de voce**

Am folosit biblioteca de text-to-speech pyttsx3 asa cum era sugerat in cerinta temei. Dupa o sesiune de experimentare am ales sa raman cu volumul si vocea default si sa modific doar rata de vorbire. Totodata am avut grija sa sincronizez si videoclipul cu sunetul.

## **Codecul video**

Codecul video asignat mie a fost DivX care este o metoda de compresie lossy bazata la inceput pe MPEG-4 Part 2 (am citit ca a inceput ca fiind o versiune hacked a codecului MPEG-4 de la Microsoft). Dupa ce am consultat lista din laboratorul 7 legata de FOURCC nu am fost complet sigur daca '3IVD' era cel mai potrivit cod asa ca m-am documentat mai mult si am descoperit materialul [1] cu o lista mai extinsa de unde am mai extras alte doua coduri care poate ar fi fost potrivite: 'DIVX' si 'DX50'. Toate cele 3 variante sunt disponibile in cod si trebuie doar comentate. Pentru crearea celui de-al doilea videoclip am folosit MoviePy care are lista de codecuri in aceasta resursa [2] si am ales varianta 'libxvid' deoarece e o implementare open-soruce a aceluiasi standard MPEG-4 Part 2 pe care e bazat si DivX.

Faptul ca aceste codecuri reprezinta o comprimare lossy se face observat prin scaderea calitatii video la al doilea clip deoarece a fost comprimat de 2 ori.

## **Referinte**

1. Headbands – Video Codecs. Website <https://www.headbands.com/gspot/videocodecs.html>
2. MoviePy – Documentation. Website <https://zulko.github.io/moviepy/ref/VideoClip/VideoClip.htm>