AI alapú történet- és képgeneráló pipeline dokumentáció

1. Bevezetés

A projekt célja egy olyan rendszer megvalósítása volt, amely több különböző mesterséges intelligencia modellt fűz össze, és ezek együttműködésével képes egy rövid történetet generálni, a történetet részekre bontani, majd az egyes részekhez illusztrációkat készíteni, végül pedig automatikus képleírást (captiont) előállítani. A rendszer működése több, egymástól eltérő AI komponensből épül fel, amelyek adatfolyamszerűen dolgoznak egymás után.

Ennek eredményeként egy olyan pipeline jött létre, amely szövegből képeket, majd képekből szöveges leírásokat készít. A projekt célja nem csupán a működés elérése volt, hanem az is, hogy mindezt egy fogyasztói szintű GPU-n, konkrétan egy **NVIDIA GTX 1660 SUPER (6GB VRAM)** kártyán keresztül is stabilan lehessen futtatni. A korlátozott erőforrások miatt a modellválasztás és a memóriakezelés kiemelten fontos részét képezte a rendszer megvalósításának.

2. Használt technológiák és környezet

2.1 Programozási környezet

A program Python nyelven készült, és több modern gépi tanulási könyvtárat használ:

- PyTorch: alapvető tensor- és GPU-kezelés
- HuggingFace Transformers: szöveg- és képleíró modellek betöltése
- Diffusers: képalkotó modellek futtatása
- huggingface_hub: lokális modell-cache és letöltés kezelése

A környezet futtatása Windows operációs rendszeren történt, CUDA 11.8 támogatással. A VRAM korlátok miatt a GPU memória kézi tisztítása is a rendszer része lett. A CUDA és PyTorch verziók összehangolása kritikus volt a stabil működéshez.

2.2 Felhasznált AI modellek

A pipeline három fő modellt használ:

Flan-T5-XL (szövegmodell)

- Feladata egy 5–6 mondatos, rövid történet generálása.
- CPU-n fut, mert VRAM-igénye meghaladja a 6 GB-ot.
- Előnye: jó koherencia rövidebb szövegekben.

Stable Diffusion v1.5 (képgenerálás)

- GPU-n fut **float32** pontossággal (FP16 instabilitás miatt).
- A safety checker ki van kapcsolva, mert VRAM-ot fogyaszt és gyakran hibás blokkolást okoz.
- 512×512 pixel felbontású illusztrációkat készít.

BLIP Image Captioning Base (képleírás)

- A képek automatikus elemzésére szolgál.
- A base modell lett választva, mert a large modell több mint 10GB VRAM-ot igényel.
- A modell **safetensors** formátumba lett konvertálva, mert a `.bin` verzió nem tölthető be PyTorch 2.5.x alatt.

2.3 Hardverkövetelmények

A rendszer fő erőforrása a GTX 1660 SUPER GPU:

- **6GB VRAM**
- SD v1.5 float32 módban körülbelül 4.5–5 GB-ot használ
- BLIP-base további 1–1.5 GB-ot igényel
- Emiatt a pipeline-ban kötelező a modellek közötti **GPU memória törlés**, különben a futás megszakadna.

A CPU oldalon legalább 16 GB RAM ajánlott, különösen a Flan-T5-XL futtatása miatt.

3. A rendszer felépítése

3.1 Könyvtárstruktúra

A projekt kimenetei mappastruktúrába rendezve kerülnek mentésre:

/output

/texts

/images

/captions

Ez a struktúra lehetővé teszi a külön komponensek eredményeinek áttekinthető elkülönítését.

3.2 A feldolgozás adatfolyama

A teljes rendszer működése négy fő szakaszra osztható:

- **Történet generálása**
  A Flan-T5-XL modell 5–6 mondatos történetet készít egy előre definiált prompt alapján.
- **Történet feldarabolása**
  A szöveg egyszerű reguláris kifejezéssel mondatokra bomlik, így minden mondat egy önálló kép alapjául szolgálhat.
- **Képek generálása mondatonként**
  A Stable Diffusion v1.5 minden mondathoz külön illusztrációt készít, 512×512 pixel méretben.
- **A képek elemzése és képleírás készítése**
  A BLIP safetensors modell minden generált képre automatikusan leírást ad, amelyet külön file-ban ment el.

4. A pipeline fő komponensei

4.1 Történetgenerálás (Flan-T5-XL)

A történetgenerálás promptalapú. A modell CPU-n fut, ami lassabb, de biztosítja, hogy a GPU szabad maradjon a képgeneráláshoz. A generált szöveg több mondatot tartalmaz, így vizuálisan jól illusztrálható.

4.2 Mondatokra bontás

A történet felbontása reguláris kifejezéssel történik:

re.split(r'(?<=[.!?]) +', story)

A cél, hogy minél több világosan elkülöníthető rész legyen, amelyhez illusztráció generálható.

4.3 Képgenerálás (Stable Diffusion v1.5)

A képgenerálás GPU-n történik:

- magas VRAM-használat miatt a modell **float32** tömböket használ
- a safety checker ki van kapcsolva
- minden prompt egy külön képet ad vissza
- a képet PNG formátumban menti el

A generálás közben a modell részmoduljai GPU-ra vannak helyezve, a generálás után pedig felszabadításra kerülnek.

4.4 BLIP Base képleírás

A BLIP képleíró modul feladata az elkészült képek automatikus értelmezése.

A base modell eredetileg *.bin* fájlt használ, ami PyTorch 2.5.1 alatt nem tölthető be a biztonsági korlátozások miatt. Ezért a modell safetensors formátumba lett konvertálva, így a betöltés stabil és gyors, valamint nem használ *torch.load()* hívást.

A captionöket külön *.txt* fájlokba menti a rendszer.

4.5 Memóriakezelés

A GPU memóriakezelés kritikus eleme a pipeline-nak, mivel a 6GB VRAM nagyon gyorsan megtelik:

- minden modell betöltése előtt:
  torch.cuda.empty_cache()
- nagy modellek törlése:
  del pipe, del model, del processor
- Python objektumok takarítása:
  gc.collect()

Ezek nélkül a Stable Diffusion és a BLIP nem tudnának egymás után futni.

5. Eredmények és tapasztalatok

A rendszer működése során több sikeres futás készült, ahol:

- a történetgenerálás megfelelő koherenciájú, többmondatos történetet adott
- a képgenerálások vizuálisan is illeszkedtek a mondatokhoz
- a BLIP által készített képleírások jól tükrözték a generált képek tartalmát

A legnagyobb kihívások a VRAM-kezeléssel, modellek kompatibilitásával és a megfelelő modellválasztással kapcsolatosak voltak. A BLIP-base modell safetensorsra konvertálása például elengedhetetlen volt a működéshez, és több hibakeresési lépés vezetett a végső megoldás megtalálásáig.

6. Összegzés

A projekt eredménye egy összetett, több AI modellt egymás után futtató rendszer lett, amely képes a teljes folyamat automatizálására: a történet megírásától a képek elkészítésén át egészen a képleírások generálásáig. A pipeline stabilan fut egy középkategóriás GPU-n, ami bizonyítja, hogy megfelelő optimalizálással nagyméretű modellek is használhatók korlátozott erőforrás mellett.

A pipeline továbbfejlesztésére lenne lehetőség, például más, korszerűbb modellek használatával:

- történetgenerálásra: **LLaMA-3 Instruct**, **Mistral-Instruct**
- képgenerálásra: **Stable Diffusion XL**, **SD-Turbo**
- képleírásra: **BLIP-2**, **LLaVA 1.6**

Ezek a modellek jobb minőségű eredményeket adhatnak, de nagyobb VRAM igénnyel működnek, így jelen projektben nem voltak alkalmazhatók.