Semestrální projekt MI-MVI 2019/2020: Video Frame Rate Upscaling Using Neural Networks

Anna Moudrá
moudrann@fit.cvut.cz

30.12.2019

Úvod

Tato práce se zabývá automatizovaným navyšováním snímkové frekvence videa interpolacemi sousledných snímků za pomoci hluboké neuronové sítě. Zvýšením počtu snímků ve videu lze dosáhnout jak videa s vyšší frekvencí snímků za vteřinu, což vede na plynulejší, přirozenější vnímání pohybu lidským okem, tak lze získat zpomalené video se stejnou frekvencí sníků za vteřinu, tedy aniž bychom museli původní snímky duplikovat. V televizním průmyslu je dosud stále ještě běžný standard 24 – 30 FPS (frames per second) nebo 50 – 60 FPS u novějších filmů. V počítačové grafice je dnes běžný standard 60+ FPS, v klasické animaci je běžnější naopak nižší obnovovací frekvence do 24 FPS, často s opakováním 2 – 4 sousledných snímků. Jelikož lidské oko je schopné vnímat obnovovací frekvenci až do 50 – 60 FPS, nad tuto hranici se jakýkoliv pohyb člověku jeví jako zcela plynulý. Cílem této práce je vyzkoušet a porovnat kvalitu dvou různých modelů, které různým způsobem interpolují mezi sousledných snímky.

Vstupní data

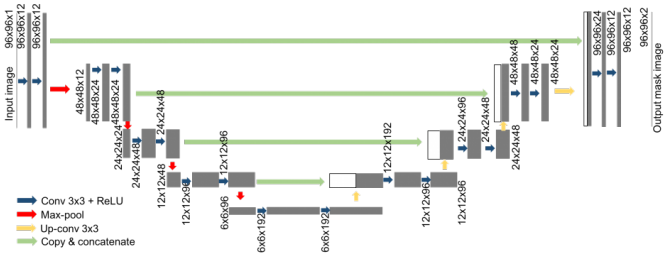
Na vstupu experimentu je vždy video ve formátu MP4 o velikost 128×384 pixelů s daným počtem snímků n. Vstupem každého konkrétního modelu je dvojice sousledných snímků x_1 a x_2. Výstupem modelu je pak snímek $x_{1,2}$ – interpolace mezi vstupními snímky. Výstupem experimentu je tedy video s $2n - 1$ snímky. Vzhledem k omezenému vnímání FPS lidským okem, byly modely testovány na videích s nízkou až hraniční obnovovací frekvencí tak, aby rozdíly mezi výstupy jednotlivých modelů byly pozorovatelné nejen na základě porovnání jednotlivých snímků, ale také celkovým subjektivním dojmem z pohybu zobrazeném ve výslednám videu.

1 Testované modely

V této práci srovnávám výsledky implementací modelu s architekturou U-Net a GAN modelu tak, jak jsou uvedeny v [5] a [4]. Oba modely jsou tedy založené na neuronových sítích s konvolučními vrstvami a liší se jak architekturou, tak přístupem vyhodnocování kvality predikce.

1.1 U-Net

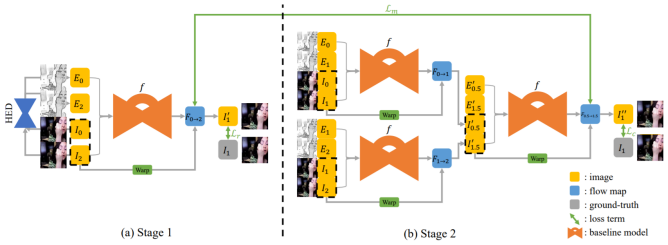
Prvním modelem je DeepMotion [5] založený na architektuře U-Net, což je typ CNN, složené z jednoho zmenšujícího (contracting via pooling layers) and z jednoho zvětšujícího (expanding via upsampling layers) bloku, často používané pro segmentaci obrazu s malým objemem dat [6]. Autor trénoval DeepMotion model za využití Charbonnierovy loss funkce na celkem 80,000 trojicích vstupních snímků KITTI datasetu [1] a následně ještě model dotrénoval na dalších cca 80,000 snímcích z datasetu YouTube-8m. Zatímco KITTI dataset jsou videa pořízená autonomními vozidly v běžném provozu, druhý dataset je velmi rozmanitý a velmi pravděpodobně obsahuje tématický přesah s naším testovacím datasetem.



Obrázek 1: Příklad architektury CNN typu U-Net.

1.2 Cyclic Frame Generation

Druhý model je založený na Generative Adversarial Network modelu, který využívají nejen klasický princip soutěžení mezi generátorem a diskriminátorem ale také tzv. cyclic consistency loss, který v přidané fázi vezme výstupy z předchozích dvou fází za vstup nového modelu a porovnává výstup tohoto nového modelu s původní ground truth tak, jak je uvedeno na obrázku 2. Tento model byl trénován na cca 28,000 snímcích datasetu UCF101 [3], což je velmi rozmanitá kolekce YouTube videí.



Obrázek 2: Příklad přidané cyklické fáze v modelu CycleGen.

2 Trénování a výběr datasetu

Jelikož oba modely jsou poměrně komplexní a vyžadují velké množství strojového času k trénování, rozhodla jsem se využít již předtrénovaných sítí dostupných na [5] a [4].

Zjevnou nevýhodou toho přístupu je, že každý model byl trénován na jiném datasetu a jinak dlouho, což bude mít jistě dopad na kvalitu výstupu modelů. Z tohoto důvodu oba modely porovnávám na rozmanitém datasetu reálných videí (tedy ne animace) s rozdílným přiblížením, osvětlením, pohybem kamery a snímanými objekty. I přestože byly převzaty již natrénované modely, jejich samotné zakomponování nebylo triviální, zejména u modelu DeepMotion kde bylo potřeba přepsat celou architekturu sítě v Keras API tak, aby bylo možné naimportovat již připravené váhy pod novými verzemi knihoven TensorFlow a NumPy. Některá použitá videa jsou má vlastní, další byla získána z veřejné knihovny Pexels [2]. Vstupní videa byla zmenšena na rozlišení 128×384 pixelů. Počet rámců videa byl podvzorkován na polovinu tak, aby bylo možné porovnávat výstupy modelů nejev vůči sobě navzájem ale i vůči originálu.

3 Hodnocení kvality modelů

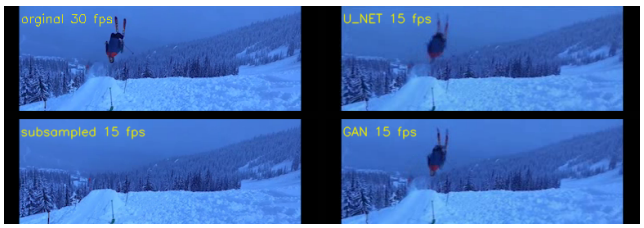
Experimenty byly provedeny na celkem 10ti různých videích a při porovnání kvality výstupů byl kladen důraz nejen na korektní zachycení větších objektů v obaze ale i na přirozenost pohybu.

3.1 Porovnání jednotlivých snímků

Na následujících snímcích je vždy zobrazen originál vlevo nahoře, vpravo nahoře je výstyp modelu DeepMotion a vpravo dole je výstup modelu CycleGen. Uvedeny zde jsou pouze 3 nejzajímavější příklady, na kterých si lze povšimnout rozdílných charakteristik obou modelů, zvláště pak jejich slabin. Na prvním videu s projíždějícím autem na obrázku 3 lze vidět, jak si oba modely snaží poradit s poměrně velkou změnou polohy objektu ve videu. Zatímco model Deep Motion interpoluje mezi dvěma snímky s větší mírou rozmazání, a tím také zbytečně vyhlazuje terén kolem silnice, model CycleGen lépe pracuje s malými změnami v obraze ale u větších objektů je velmi patrný efekt zdvojení, který subjektivně působí ve statickém snímku velmi nepřirozeně. Na snímku 4 s lyžařem lze pozorovat jak

Obrázek 3: Export snímku

modely nakládají s pohybem objektu na relativně konstantní pozici ve snímku bez výrazného pohybu kamery. DeepMotion tentokrát nevyhlazuje pozadí a správně interpoluje jen část snímku s lyžařem, naopak slabinou modelu je ztráta detailu a obrysů původního objektu. Výstup modelu CycleGen je v podstatě jen mírně rozmazanější verze skutečného chybějícího snímku pouze s mírnou změnou úhlu lyžare, což je zaviněno nelinearitou snímaného pohybu. Na posledním snímku 5 běžícího stáda je zachycen nejen pohyb mnoha objektů napříč snímkem ale i nestabilní vertikální pohyb kamery. Stejně jako v předchozích příkladech model DeepMotion má problémy zachovat detaily, k tomu se přidává vertikální i horizontální pohyb kamery, což vede na velké zašumění celého výstupního snímku. Model CycleGen daleko lépe nakládá s pohybem kamery a přestože je snímek rozmazanější než originál, stále je rozpoznatelná velká míra detailů jako jsou jednotlivé větve nebo paroží. Nevýhodou jsou velké duplicity rychle se pohybujících objektů stejně jako v případě prvního snímku, tedy na snímku je zobrazeno daleko více nohou než v originále.



Obrázek 4: Export snímku

Obrázek 5: Export snímku

3.2 Porovnání výsledných videí

Videa jednotlivých výstupů byla zkompletována a jsou k nahlédnutí na GitLab stránce tohoto projektu. Oba

modely velmi dobře pracují s pomalým plynulým pohybem zachyceným na statickou kameru. Obecně také platilo, že oba modely přinášejí do videa výrazný šum. Subjektivně hodnotím CycleGen jako kvalitnější model, který je schopen pracovat i s jemnějšími detaily v obraze s nízkým rozlišením.

4 Závěr

Z provedených experimentů vyplynulo, že CNN jsou vhodným prostředkem pro interpolaci snímků za účelem zvýšení snímkové frekvence videa. Oba modely fungují dobře na videa pořízená statickou kamerou, naopak problémy oběma modelům dělají již příliš podvzorkovaná vstupní data. Lze tedy logicky předpokládat, že čím vyšší bude FPS frekvence vstupních dat, tím lépe budou vypadat i generované výsledky. Pokud nám tedy nevadí přidaný šum a drobné defekty, lze model teoreticky aplikovat dvakrát i víckrát a získat tak několikanásobnou frekvenci FPS. Bohužel při opětovné aplikaci na data s velmi nízkou počáteční frekvencí kvalita videa rychle degeneruje, proto pokládám oba modely pro toto použití nevhodné. Pro lepší porovnání skutečné kvality výstupů a případných možností komerčního využití by bylo nutné modely otestovat, v případě DeepMotion tedy i odstranit normalizační vrstvy a přetrénovat, na datasetech s vyšším rozlišením.

Reference

[1] Kitti vision benchmark dataset. Dostupné z http://www.cvlibs.net/datasets/kitti/.

[2] Pexels videos. Dostupné z https://www.pexels.com/videos/.

[3] Ucf101 - action recognition data set. Dostupné z https://www.crcv.ucf.edu/data/UCF101.php.

[4] Yu-Lun Liu, Yi-Tung Liao, Yen-Yu Lin, and Yung-Yu Chuang. Deep video frame interpolation using cyclic frame generation, 2019. Dostupné z https://github.com/alex04072000/CyclicGen.

[5] Duncan Woodbury Neil Joshi. Deep motion:a convolutional neural network for frame interpolation, 2017. Dostupné z https://github.com/neil454/deepmotion.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. Dostupné z https://arxiv.org/abs/1505.04597.