

KÉPÉLESÍTÉS

“Super resolution” algoritmus használata

RÖVID ÖSSZEFOGLALÁS

A képek élesítése és nagyítása egy megszokott jelenet ma már számos filmben. Ezt próbálom imitálni “szuper felbontás” algoritmus használatával kép élesítésével és esetleges nagyításával ahol szükséges.

Burian Sándor, AWXYHE

Számítógépes képfeldolgozás és grafika
(NIXSKGHMNE)

2020-2021, második félév

Óbudai Egyetem, Neumann János Informatikai kar

Tartalom

Alapötlet	2
Kifejtés	2
Super resolution algoritmusok néhány típusa	3
Jóslás alapú modellek.....	3
Éldetektálással	3
Statisztikai módszerek	3
Tanulási módszerrel	3
A használt módszerről részletesen	4
CNN VS ResNet VS GAN	4
Részletesebben a ResNet működéséről	5
Tanítási adathalmaz	5
Megvalósítás TensorFlow használatával	6
A használt módszer elmélete	6
A használt módszer gyakorlatban	7
Eredmények minta képekre:	8
Eredmények saját képekre:.....	9
Képjegyzék	10
Forrás	11

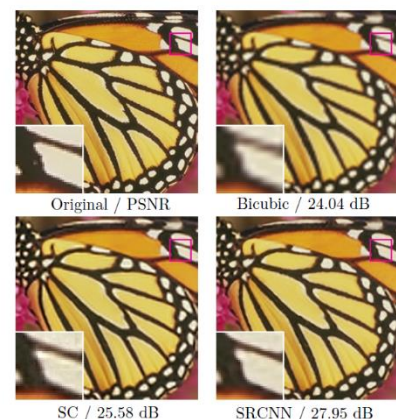
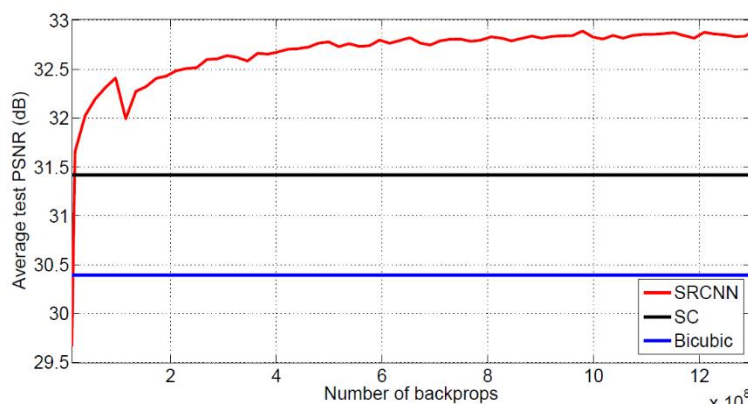
Alapötlet

Van egy képünk amit élesíteni szeretnénk, vagy csak egy részletét a képnek, de ebben az esetben az a részlet lehet kis felbontású is, ezért akár növelésre is szükség lehet, olyan mértékben amennyire csak tudjuk veszteségmentesen. Ezt a jelenleg használatos mélytanulási technológiákkal lehetővé teszik, hogy bizonyos esetekben akár 8-szorosára növeljük a kép méretét jelentős minőségvesztés nélkül (Anwar, 2020)[1]! Ezt szeretnénk elvégezni pár képen.

Kifejtés

Tehát gyakorlatilag akár képrészletről akár az egész képről beszélünk egy kép méret növelés a célunk még hozzá minőségromlás nélkül [1]. Ellenpéldaként viszont nem árt röviden tisztázni, hogy a legtöbb algoritmus ami a kép méretének növelését szolgálja az hogy működik. Ilyen a legközelebbi szomszéd és bilineáris interpoláció, előbbi széthúzza a képet, majd a keletkező réseket a legközelebbi pixel színére tölti fel, ha a környező 16 pixel átlagát vesszük valamivel jobb a végeredmény, ez a bicubic, avagy kettős köbös interpoláció; utóbbi, a kettős lineáris módszer függőleges és vízszintes irányban is értékeket vesz amik között fokozatosan csökkenti az átmenetet, ennek megfelelően egy jelentősen elmosódott képet kapunk eredményként (Petur, 2005)[2].

Nos mint ahogy azt sem a fentebb mutatott eljárások nem teszik lehetővé, nem tudunk csodát tenni, a nem ismert képpontokat nem tudjuk meghatározni teljes pontossággal, de jobban tudjuk közelíteni, ha “emlékszünk rá”, hogy hasonló esetekben mi történt. Alább a különböző kép méret növelése közben használt algoritmusok hatásai látszanak. A jobb alsó mutatja az eredményét annak a technológiának amit alább használni fogunk.



kép 1 Dong , Loy, He, Tang - Image Super-Resolution Using Deep Convolutional Networks

Az alapötlet egyszerű: mutassuk meg egy neurális hálónak a képeket és azok lebutított, felbontás csökkentett változatait, majd várjunk egy nagyobb felbontású képet kimenetként amit az eredeti képpel összehasonlítunk.

A valóságban ellenben kicsivel bonyolultabb a helyzet, érdemes lehet ugyanis színskálára bontatni az egész művelet előtt majd utána újra egyesíteni, ugyanis kísérletek bizonyítják, hogy a végeredmény így jobb lesz (Dong et al., 2015)[3].

Super resolution algoritmusok néhány típusa

Érdemes figyelembe venni, hogy nem csupán egy megközeítése létezik ennek a problémának. Létezik várható érték, éldetektálás, kép statsztikai és példa alapú megközelítések is. Alább rövid áttekintése párnak, csupán inspirációs jelleggel (Yang, Ma, Yang, 2014)[4].

Jóslás alapú modellek

Mélytanulás, vagy bármilyen tanulási módszer nélkülözésével, csupán egy előre meghatározott matematikai formula segítségével próbál magasabb felbontást elérni. Tipikusan ezek a bilineáris, bicubic vagy a Lánczos formulák. Mivel ezek mind a szomszédos pixelekhez viszonyítanak, így nem kérdés, hogy lokálisan ugyan elfogadható a képminőség javulása, de az éleket nem veszik figyelembe, a képek szélénél szintén bajban vannak, és a teljes összkép is elmarad a várttól

Éldetektálással

Az éldetektálás használatával jelentősen javíthatunk a köztes régiók minőségén, de az élek markánsak lesznek a végeredményen, illetve például textúrák tekintetében alul marad az elvárhoz képest. Ezek okán érdemes lehet például szövegekhez, vagy hasonló írásjelek és hasonló kiemeléséhez, de általános képekhez, amiken bármi szerepelhet nem lesz kielégítő.

Statsztikai módszerek

Különböző képtulajdonságokat használatával különböző statsztikai eszközt használhatunk, gradinesek súlyozásával, eloszlásával, de ezek nem feltétlen elvárhatóak egy általános képbemenetnél, így a módszer maga szintén korlátozott területen használható, ugyakkor jelentős erőfárrás igénye is van.

Tanulási módszerrel

Ahogy fentebb is részleteztem, külső kép adatbázis szükséges hozzá, ahol ugyanannak a képnek mind a nagy mind az alacsony felbontású verziója létezik. Erre többféle tanulási módszert is használhatunk, támogatott és mélytanulást is. Azon kívül több módszert használhatunk az átfedésben lévő területek fledolgozására is.

Ezek képességének összehasonlítására született tanulmány[4], melyben kifejezetten a nem kategorizált képeket vették alapul, hogy ne egy-egy képtípusra

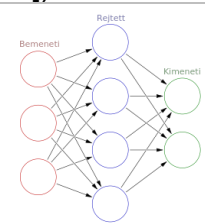
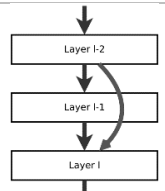
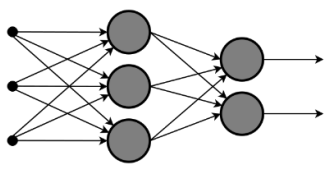
legyenek érvényesek a mérések, és ezek alapján a mélytanulós módszert érdemes előtérbe helyezni, még egyszer szeretném kiemelni, hogy csak általános képek esetén, specifikusan, adott képtípusokra ez már nem igaz!

A használt módszerről részletesen

Hasonló módszerrel dolgozik a Google, Nvidia, Pixelmator, Adobe Photoshop is, igaz ők saját tanulási algoritmusokat fejlesztenek alá, de az alapötlet, a kis és nagyfelbontású képek használatáról nem változott (Vincent, 2019)[5]. A Google a Pixel 3 telefonjaiban már úgy építette be a technológiát, hogy a közléítés, „zoomolás” maga a kamera applikációban egy Super Resolution Zoom. Így szembementek azzal az elvvel, hogy telefonon az optikai zoom hiányában értelmetlen a zoom. Sőt, a felhasználó is jobban járt ha ráközelített, mintha utólag vágta volna körbe a kijelölt részt. Ugyanakkor ők nem kész képet dolgoztak fel, hanem HDR képből indultak ki, tehát alpból több kép készült közel azonos pozícióból, amit összevetve alpból a 2-3x optikai zoommal egyenértékű közelítést lehet elérni a képen (Wronski, 2018/2020)[6]. De mivel itt már kész, képekről beszélünk, amik valamikor elkészültek ezért az az út nem járható ebben az esetben, ugyanakkor mindenképp figyelemre méltó elgondolás lehet más esetekben, például élő felvételen való alkalmazáshoz.

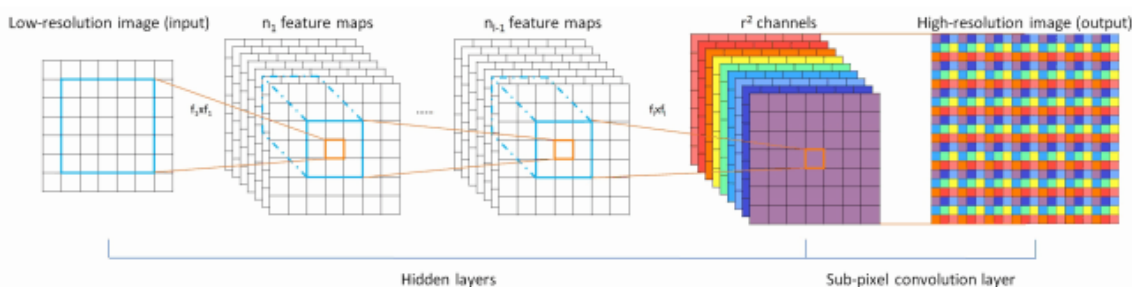
CNN VS ResNet VS GAN

Sok megoldás (pl: Shi et al., 2016[7]; [3]; Thomas, 2019[8]) konvolúciós hálózatot használ (CNN), vagy megmaradó neurális hálót (ResNet) (pl: Zhang et al., 2018[9]), esetleg generatív modelleket (GAN) (pl: Wang et al., 2018[10]; Ledig et al., 2017[11];), és van olyan megoldás ami keveri, pontosabban szólva felhasználja ezeket (pl: Cardinale, 2020[12]). Ezért szükségesnek érzem, hogy a köztük lévő különbségről is essen pár szó.

	CNN	ResNet	GAN
Technológiáról	Ideális képek feldolgozásához, a képeken filtereket alkalmazunk, majd a bemenet és a hálózat hibáinak kijavítása után osztályozzuk a kapott eredményeket (szaboa, 2018 [13])	A CNN-nél tapasztaltak felhasználásával, annak továbbfejlesztésével készült, hogy a konvolúciós rétegben blokkokat kezeljen[7]. Az eredmény jelentősen jobb mint a CNN esetén.	„a tanítóadatokhoz hasonló, új adatokat szeretnénk létrehozni. Ennek egyik példája a lehetséges jövő szimulálása” (Csapó, [14])
Ábrázolása	 <p>kép 2 - CNN működése</p>	 <p>kép 3 - Canonical form of residual neural nets</p>	 <p>kép 4 - A neural network with multiple layers</p>

Ezek alapján elmondhatjuk, hogy a ResNet és a GAN használata lehet a legkifizetődőbb a mi szempontunkból, én mivel részletesebb irodalmat találtam a ezekről, [1][8] így ezekkel fogok dolgozni a következőkben.

Részletesebben a ResNet működéséről



kép 5 - a1 pixelek használata (ESPCN) - [7]

Mivel az átméretezéskor keletkező köztes, eddig nem létező pixelekről nincs információnk, ezért ez a módszer egy köztes tanulási réteget alkalmaz ezen a ponton, ami növelni tudja a kép értékét. A kép értékét a következő képlet adja meg:

$$PSNR = 10 \cdot \log_{10} \frac{MAX_I^2}{MSE} \quad (1)$$

Ahol:

PSNR – csúcsérítkek zajhoz vett aránya

MAX_I^2 – maximum pixelszám a képrészleten

MSE – zaj mértéke

Azaz, ha csökkentjük a bemeneti zajt, a képrészleten, akkor növelni tudjuk a végső kimenet értékét. Ez a köztes réteg ezt is végzi el nekünk. Ugyanakkor fontos megemlítenünk, hogy ha túl nagy mértékben használunk elmosást akkor a végeredmény bár matematikailag jobb közelítést ad, de az emberi szemnek nem tűnik optimálisnak. Ezen a ponton lehet jelentős számunkra a GAN technológia, mivel itt gyakorlatilag létrehozunk képpontokat, a korábbi statisztikák alapján, ami természetesen nem fog teljesen megegyezni a korábbi nagy felbontású képnél látott képpontokkal, ugyanakkor elég jól elhitetik azt velünk, hogy olyan hasonlít rá, hogy az már elegendő legyen számunkra.

Tanítási adathalmaz

Tanításhoz használható lehet a DIV2Kⁱ adathalmaz 800 képe, az imagenetⁱⁱ 1,2 millió képe vagy saját gyűjtés is szóba kerülhet szabadon, CC licenz alatt futó adatbázisokból, pl pixabay.com-ról.

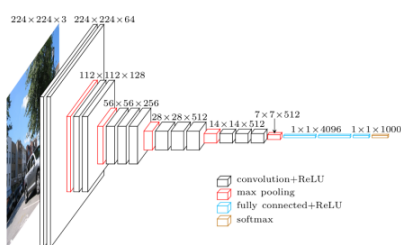
ⁱ <https://data.vision.ee.ethz.ch/cvl/DIV2K/> elérhető: 2021 03 26

ⁱⁱ <http://www.image-net.org/> elérhető 2021 03 27

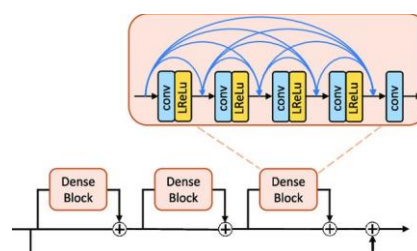
Megvalósítás TensorFlow használatával

A használt módszer elmélete

A használt eljárásban előbb módosításokat kell végezni a korábban használt eljárásokhoz képest. Az egyik ilyen ahogy Wang[10] is írja egy megamaradó réteg alkalmazása. Pontosabban több ilyen alkalmazása, a köztes ugrások engedélyezésével, ahogy az a (ábra 7) látható. Ez gyorsítani fog a hálózaton és megkönnyíti a tanítást is. Ugyanakkor szükségételenné teszi a köteget normalizálást. Továbbá egy RaGAN réteg használatára is szükségünk lesz ami azt hivatott eldönteni, hogy „az egyik kép realisztikusabb-e mint a másik”[10]. Ugyanakkor még elé szükséges beilleszteni egy réteget ami egyfelől a mélytanulás miatt szükséges, másfelől a nagyfelbontású képek használata miatt, ez a VGG, (ábra 6).

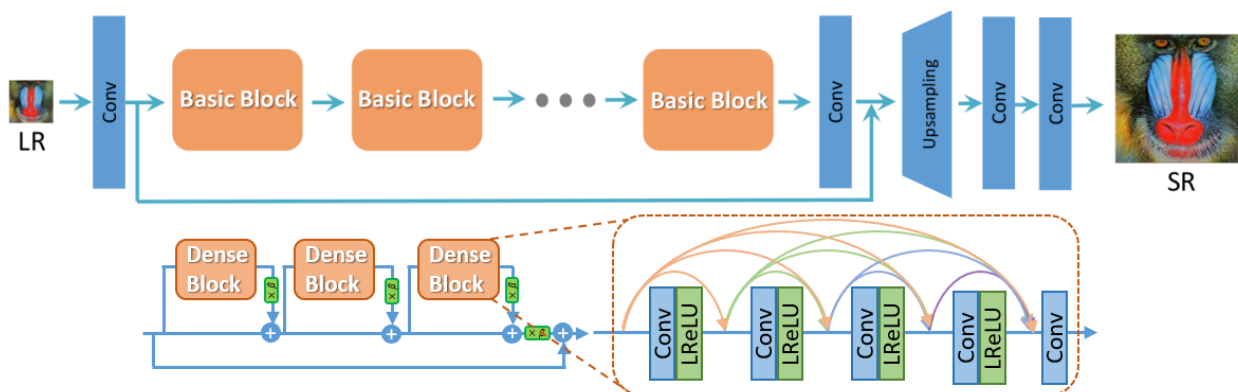


kép 6 - Macroarchitecture of VGG16



kép 7 - The residual in residual dense block (RRDB)

Ennek a módszernek nagy előnye, hogy a kis felbontású képen hajtjuk végre a módosításokat, sőt még gyorsíthatunk is, ha alapvetőbb egységeket hozunk létre, minden részfeladatnak külön-külön, ugyanakkor a korábbi alablokkokat kicseréljük RRDB (ábra 7) blokkokra, ez a módszer a feladat általánosításában is segít, így nem kell külön-külön témákra betanított hálózatot használnunk, valamint segíti a szükséges számítási kapacitás csökkentését és használt memória méretének csökkentését is. Így kapjuk a hálózatot amit (ábra 8) mutat.



kép 8 - We employ the basic architecture of SRResNet, where most computation is done in the LR feature space. We could select or design “basic blocks”(e.g., residual block, dense block, RRDB) for better performance – [10]

Mindez azon is alapul, hogy a több egymás utáni RRDB blokk eredményesebb összességében mintha „csak” SRGAN-t használnánk.

Ahhoz, hogy az eljárás okozta nagy mértékű zajt csökkentsük először egy már korábban bemutatott, PSNR-t figyelembe tartó, arra fókuszáló hálózaton tanítjuk, ezután alkalmazunk GAN hálózatot, a fonomításra. Ennek az az értelme, hogy a modell elsősorban képes legyen egy minőségi javításra de finomítható legyen, és finomodjon újratanítás nélkül is!

A használt módszer gyakorlatban

A megvalósításhoz Dey, 2019[15] munkáját vettem alapul amihez Wang[10] leírását használtam alapként. Tehát a TensorFlow segédkönyvtár segítségével a esrgan-tf2 (Dey, 2019[16]) módszert használtam, ami egy előtanított hálót tartalmaz. Ennek előfeltétele, hogy a képünk egy float32 típusú kép legyen, és alfa csatorna nincs támogatva. Tehát a képet éát kellett alakítani, hogy biztosan megfeleljen a bemenetnek:

```
1. def preprocess_image(image_path):
2.     hr_image = tf.image.decode_image(tf.io.read_file(image_path))
3.
4.     if hr_image.shape[-1] == 4:
5.         hr_image = hr_image[..., :-1]
6.     hr_size = (tf.convert_to_tensor(hr_image.shape[:-1]) // 4) * 4
7.     hr_image = tf.image.crop_to_bounding_box(hr_image, 0, 0, hr_size[0], hr_size[1])
8.     hr_image = tf.cast(hr_image, tf.float32)
9.     return tf.expand_dims(hr_image, 0)
```

Kódrészlet 1 – kép előformázása

Ezt követően ahogy am intakód is mutatja tudjuk alkalmazni a képre az eljárást:

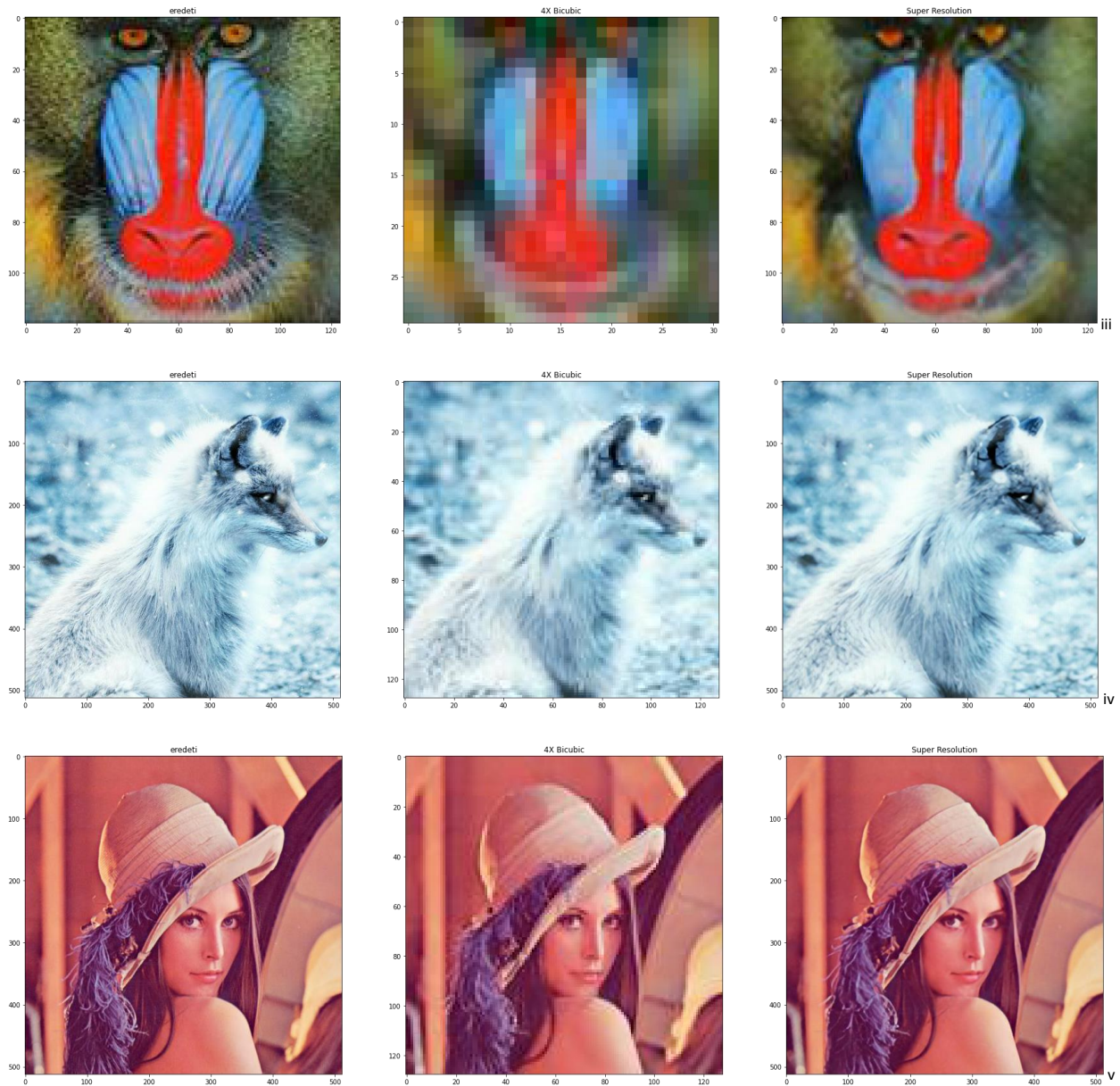
```
1. import tensorflow_hub as hub
2. import tensorflow as tf
3.
4. model = hub.load("https://tfhub.dev/captain-pool/esrgan-tf2/1")
5. low_resolution_image = load_image()
6. low_resolution_image = tf.cast(low_resolution_image, tf.float32)
7. super_resolution = model(low_resolution_image)
```

Kódrészlet 2 – Super resolution alkalmazása képre, forrás: <https://tfhub.dev/captain-pool/esrgan-tf2/1>

A teljes saját alkalmazás, és összehasonlítás más eljárással a következő linken érhető el: (2021 05 10):

<https://colab.research.google.com/drive/1-v0eLZXN9aJbYEhewYufErOIC5P3Ufg4?usp=sharing>

Eredmények minta képekre:



iii <https://user-images.githubusercontent.com/12981474/40157448-eff91f06-5953-11e8-9a37-f6b5693fa03f.png> (2021 05 08)

iv <https://lh4.googleusercontent.com/-Anmw5df4gj0/AAAAAAAAAI/AAAAAAAAAc/6HxU8XFLnQE/photo.jpg64> (2021 05 08)

v [https://en.wikipedia.org/wiki/Lenna#/media/File:Lenna_\(test_image\).png](https://en.wikipedia.org/wiki/Lenna#/media/File:Lenna_(test_image).png) (2021 05 10)



viii

ix

X

- 9

Képjegyzék

Borítókép: 2020 Március 6, forrás: digiretus.com: <http://digiretus.com/wp-content/uploads/2016/12/photoshop-tutorial-2-sharpening-l-4.jpg>

1. Kép;2.old - C. Dong, C. C. Loy, K. He, X. Tang, (2015 07 31) *Image Super-Resolution Using Deep Convolutional Networks*. Elérhető: (2021 03 24), forrás: cuhk.edu.hk: <http://mmlab.ie.cuhk.edu.hk/projects/SRCNN.html>
2. Kép;4.old – Csxeba (2017 12 09), *Szines neuralis halo*, elérhető (2021 03 27), forrás: wikipédia: https://hu.wikipedia.org/wiki/F%C3%A1jl:Szines_neuralis_halo.png
3. Kép;4.old – Jabled (2017 11 24), *Canonical form of residual neural nets*, elérhető (2021 03 27), forrás: wikipédia: <https://en.wikipedia.org/wiki/File:ResNets.svg>
4. Kép;4.old - Offnfopt (2015 04 12), *A neural network with multiple layers*, elérhető (2021 03 27), forrás wikipédia: https://en.wikipedia.org/wiki/Generative_adversarial_network#/media/File:Multi-Layer_Neural_Network-Vector-Blank.svg
5. Kép;5.old – W. Shi, et al., 2019 – [7], *The proposed efficient sub-pixel convolutional neural network (ESPCN), with two convolution layers for feature maps extraction, and a sub-pixel convolution layer that aggregates the feature maps from LR space and builds the SR image in a single step.*, Elérhető (2021 03 26), forrás arXiv: <https://arxiv.org/pdf/1609.05158.pdf>
6. Kép;6.old D. Frossard (2016 06 17), *Fig2 Macroarchitecture of VGG16* elérhető(2021 05 10), forrás: toronto.edu: <https://www.cs.toronto.edu/~frossard/post/vgg16>
7. Kép;6.old H. Wang, D. Su, C. Liu, L. Jin, X. Sun, X. Peng – (2019 12 2) *Deformable Non-Local Networkfor Video Super-Resolution -Fig5.*, elérhető(2021 05 10), forrás: researchgate: https://www.researchgate.net/publication/337997289_Deformable_Non-Local_Network_for_Video_Super-Resolution
8. Kép;6.old Wang Xintao et al. 2018 – [10], *ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks*, elérhető (2021 05 10), forrás: readthedocs.io: https://esrgan.readthedocs.io/en/latest/_images/architecture.png

Forrás

- 1 T. Anwar, (2020 03 28) *Super Resolution with OpenCV*. Elérhető:(2021 03. 11.) , forrás: bleedai.com: <https://bleedai.com/super-resolution-with-opencv/>
- 2 “Petur”, (2005 09 27) *Interpolációs eljárások*. Elérhető:(2021 03 24), forrás: pixinfo.com: <https://pixinfo.com/cikkek/interpolacios-eljarasok/>
- 3 C. Dong, C. C. Loy, K. He, X. Tang, (2015 07 31) *Image Super-Resolution Using Deep Convolutional Networks*. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Preprint, vol. 38, no. 2, pp. 295-307, 1 Feb. 2016, doi: 10.1109/TPAMI.2015.2439281.. Forrás: <https://ieeexplore.ieee.org/document/7115171?arnumber=7115171>
- 4 Yang CY., Ma C., Yang MH. (2014) *Single-Image Super-Resolution: A Benchmark*. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8692. Springer, Cham. https://doi.org/10.1007/978-3-319-10593-2_25
- 5 J. Vincent, *AI super resolution lets you ‘zoom and enhance’ in Pixelmator Pro*. Elérhető: (2021 03 24), forrás: The Verge: <https://www.theverge.com/2019/12/17/21025811/ai-super-resolution-zoom-enhance-pixelmator-pro>
- 6 B. Wronski, (2018 09 15; frissítve 2020-ban), *See Better and Further with Super Res Zoom on the Pixel 3*, Elérhető: (2021 03 11), forrás: googleblog: <https://ai.googleblog.com/2018/10/see-better-and-further-with-super-res.html>
- 7 W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, Z. Wang, (2016 09 23), *Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network*. Elérhető (2021 03 26), forrás arXiv: <https://arxiv.org/pdf/1609.05158.pdf>
- 8 C. Thomas, (2019 02 24), *Deep learning based super resolution, without using a GAN*. Elérhető (2021 03 26), forrás: towards data science: <https://towardsdatascience.com/deep-learning-based-super-resolution-without-using-a-gan-11c9bb5b6cd5>
- 9 Y. Zhang, Y. Tian, Y. Kong, B. Zhong, Y. Fu (2018 03 27), *Residual Dense Network for Image Super-Resolution*, elérhető (2021 03 27), forrás: arXiv: <https://arxiv.org/pdf/1802.08797.pdf>
- 10 X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, Chao Dong , C. C. Loy, Y. Qiao, X. Tang (2018 09), *ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks*, Elérhető: (2018 09 17), forrás: arXiv: <https://arxiv.org/pdf/1809.00219.pdf>
- 11 C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, (2017 04 25), *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*, Elérhető: (2021 03 27), forrás: arXiv: <https://arxiv.org/pdf/1609.04802.pdf>
- 12 F. Cardinale, Z. John, D. Tran, (2020 01 08), *Image Super-Resolution (ISR)*, Elérhető (2021 03 11), forrás: github: <https://idealo.github.io/image-super-resolution/>
- 13 Szabó A. (2018 11 17), *Convolutional Neural Network felépítése dióhéjban*, Elérhető (2021 03 27), forrás: dotnetszeged: <http://www.dotnetszeged.hu/mic/?p=1750>
- 14 Csapó T. G., *GAN - Versengő hálózatok és generatív modellek deep learning-ben*, Elérhető (2021 03 26), forrás: bme: <https://iw.tmit.bme.hu/education/studenttopic/TMIT2018-139>
- 15 A. Dey, 2019, *Image Super Resolution using ESRGAN*, Elérhető (2021 05 09), forrás: tensorflow.org: https://www.tensorflow.org/hub/tutorials/image_enhancing
- 16 A. Dey, (2021 05 06), *esrgan-tf2*, Elérhető (2021 05 09), forrás: TensorFlow Hub: <https://tfhub.dev/captain-pool/esrgan-tf2/1>