

POLITECHNIKA WARSZAWSKA

WYDZIAŁ ELEKTRYCZNY

KIERUNEK INFORMATYKA

**Wykorzystanie głębokich sieci
neuronowych do poprawy rozdzielczości
zdjęć twarzy.**

Wykonał:
Aliaksandr KAROLIK

Promotor:
dr inż. Grzegorz SARWAS

6 maja 2020



Spis treści

1	Cel projektu	2
2	Wstęp	2
3	Wstęp teoretyczny	2
4	Architektury dla SISR	4
4.1	SRCNN	4
4.1.1	Wyniki działania algorytmu	6
4.2	EDSR	6
4.2.1	Wyniki działania algorytmu	6

1 Cel projektu

Praca skupiała się na badaniu najnowszych rozwiązań algorytmicznych w dziedzinie widzenia komputerowego służących do poprawy rozdzielczości, zwanych również algorytmami super-rozdzielczości (super-resolution). Wybrane metody rokujące swoją użyteczność w przypadku poprawy zdjęć twarzy zostali zaimplementowane i przebadane.

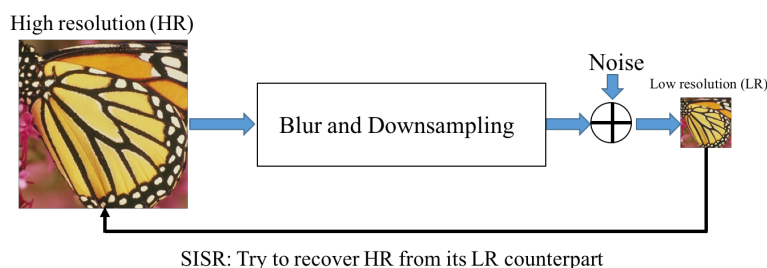
2 Wstęp

Czym jest super-rozdzielczość? Super-rozdzielczość (pisana również jako super resolution, superresolution) jest określeniem zestawu metod zwiększania skali wideo lub obrazów. Terminy takie jak „skalowanie w górę”, „powiększanie”, „konwersja w górę” i „uprez” również opisują wzrost rozdzielczości w przetwarzaniu obrazu lub edycji wideo. Większość technik super-rozdzielczości opiera się na tym samym pomysle: wykorzystanie informacji z kilku różnych obrazów do stworzenia jednego powiększonego obrazu. Algorytmy próbują wyodrębnić szczegóły z każdego obrazu w sekwencji, aby zrekonstruować inne ramki.

Głównym celem super-rozdzielczości jest wygenerowanie obrazu o wyższej rozdzielczości z obrazów o niższej rozdzielczości. Obraz w wysokiej rozdzielczości oferuje dużą gęstość pikseli, a tym samym więcej szczegółów na temat oryginalnej sceny. Potrzeba wysokiej rozdzielczości jest powszechna w wizji komputerowej aplikacji dla lepszej wydajności w rozpoznawaniu wzorów i analizie obrazów. Wysoka rozdzielczość ma znaczenie w obrazowaniu medycznym dla diagnozy. Wiele aplikacji wymaga powiększenia określonego obszaru zainteresowania obrazu, w którym niezbędna jest wysoka rozdzielczość, np. aplikacje do nadzoru, kryminalistyki i obrazowania satelitarnego.

3 Wstęp teoretyczny

Super-rozdzielczość (SR) odnosi się do zadania przywracania obrazów o wysokiej rozdzielczości z jednej lub więcej obserwacji tej samej sceny w niskiej rozdzielczości (LR). Zgodnie z liczbą wejściowych obrazów LR, SR można podzielić na super-rozdzielczość pojedynczego obrazu (SISR) i super-rozdzielczość wielu obrazów (MISR). W porównaniu z MISR, SISR jest znacznie bardziej popularny ze względu na wysoką wydajność. Typowa struktura SISR, wygląda następująco:



Rysunek 1: Szkic SISR

Głównie algorytmy SISR dzielą się na trzy kategorie: metody oparte na interpolacji, metody oparte na rekonstrukcji oraz metody oparte na uczeniu. Metody SISR oparte na interpolacji, takie jak interpolacja dwusześcienna (bicubic interpolation) i próbkowanie Lanczosa (Lanczos resampling), są bardzo szybkie i proste, ale dość nie dokładne.

Metody SR oparte na rekonstrukcji, często przyjmują zaawansowaną wcześniejszą wiedzę w celu ograniczenia możliwej przestrzeni rozwiązań z korzyścią polegającą na generowaniu elastycznych i ostrych szczegółów. Jednak wydajność wielu metod opartych na rekonstrukcji szybko spada, gdy zwiększa się skala, oraz metody te są zwykle czasochłonne.

Metody SISR oparte na uczeniu, znane również jako metody oparte na przykładach, najczęściej używane ze względu na ich szybkie obliczenia i wyjątkową wydajność. Metody te zwykle wykorzystują algorytmy uczenia maszynowego do analizy związków statystycznych między LR i odpowiadającym mu odpowiednikiem HR z istotnych przykładów szkoleniowych.

Technika MISR wykorzystuje jako wejście zestaw obrazów niskiej rozdzielczości do budowy obrazu HR, ale jak już wcześniej było wspomniane, SISR jest popularniejsza ze względu na wysoką wydajność.

4 Architektury dla SISR

4.1 SRCNN

SRCNN(Image Super-Resolution Using Deep Convolutional Networks). Architektura SRCNN pokazana jest na rysunku 2. Jak ustalono w wielu tradycyjnych metodach, dla uproszczenia SRCNN używa tylko komponenty luminancji do treningu. SRCNN jest trójwarstwowym CNN (Konwolucyjne sieci neuronowe), w którym znajdują się rozmiary filtrów każdej warstwy $64 \times 1 \times 9 \times 9$, $1 \times 32 \times 5 \times 5$ i $1 \times 32 \times 5 \times 5$. Dalej w omówieniu działania algorytmu będę stosować następującą notację:

- Y obraz o niskiej rozdzielczości
- X prawdziwy obraz o wysokiej rozdzielczości

Każda warstwa odpowiada za następujące czynności:

1. Wyodrębnienie i reprezentacja. Popularną strategią w rekonstrukcji obrazu jest wyodrębnienie plastrów, a następnie reprezentowanie ich przez zestaw wstępnie przeszkolonych baz, takich jak PCA. Jest to równoważne z zawijaniem obrazu przez zestaw filtrów, z których każdy jest podstawą. W naszym sformułowaniu, włączamy optymalizację tych baz do optymalizacji sieci. Formalnie, nasza pierwsza warstwa wyrażona jest jako operacja:

$$F_1(Y) = \max(0, W_1 * Y + B_1)$$

Gdzie:

- W_1 - odpowiedni filter
 - B_1 - szum
 - $*$ - oznacza operację zwijania (convolution operation)
2. Mapowanie nieliniowe. Pierwsza warstwa wyodrębnia $n1$ -wymiarową cechę dla każdego plastra. W drugiej operacji mapujemy każdy z tych $n1$ -wymiarowych wektorów na $n2$ -wymiarowy. Jest to równoznaczne z zastosowaniem $n2$ filtrów. Działanie drugiej warstwy jest następujące:

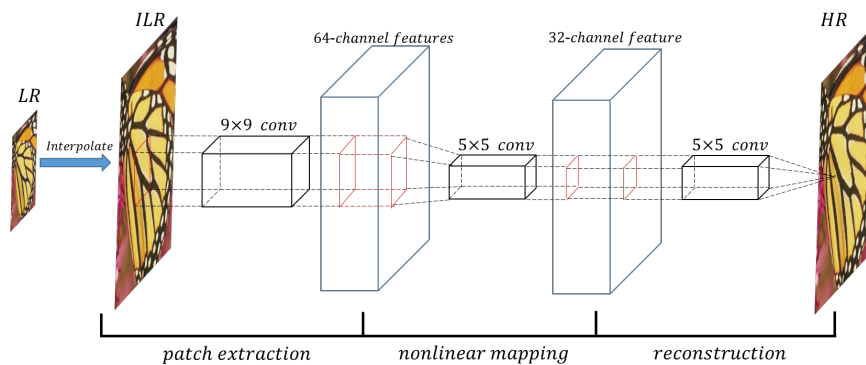
$$F_2(Y) = \max(0, W_2 * F_1(Y) + B_2)$$

- W_2 - odpowiedni filter
- B_2 - szum
- $*$ - oznacza operację zwijania (convolution operation)

Każdy z wyjściowych $n2$ -wymiarowych wektorów jest koncepcyjnie odwzorowaniem plastra o wysokiej rozdzielczości, który zostanie użyty do rekonstrukcji.

3. Rekonstrukcja. Ostatecznie plastry o wysokiej rozdzielczości są uśredniane w celu uzyskania ostatecznego pełnego obrazu. Uśrednianie może być traktowane jako predefiniowany filtr na zestawie map cech (gdzie każda pozycja jest spłaszczoną formą wektorową plastry o wysokiej rozdzielczości). Ostatnia warstwa konwolucyjną jest definiowana następująco:

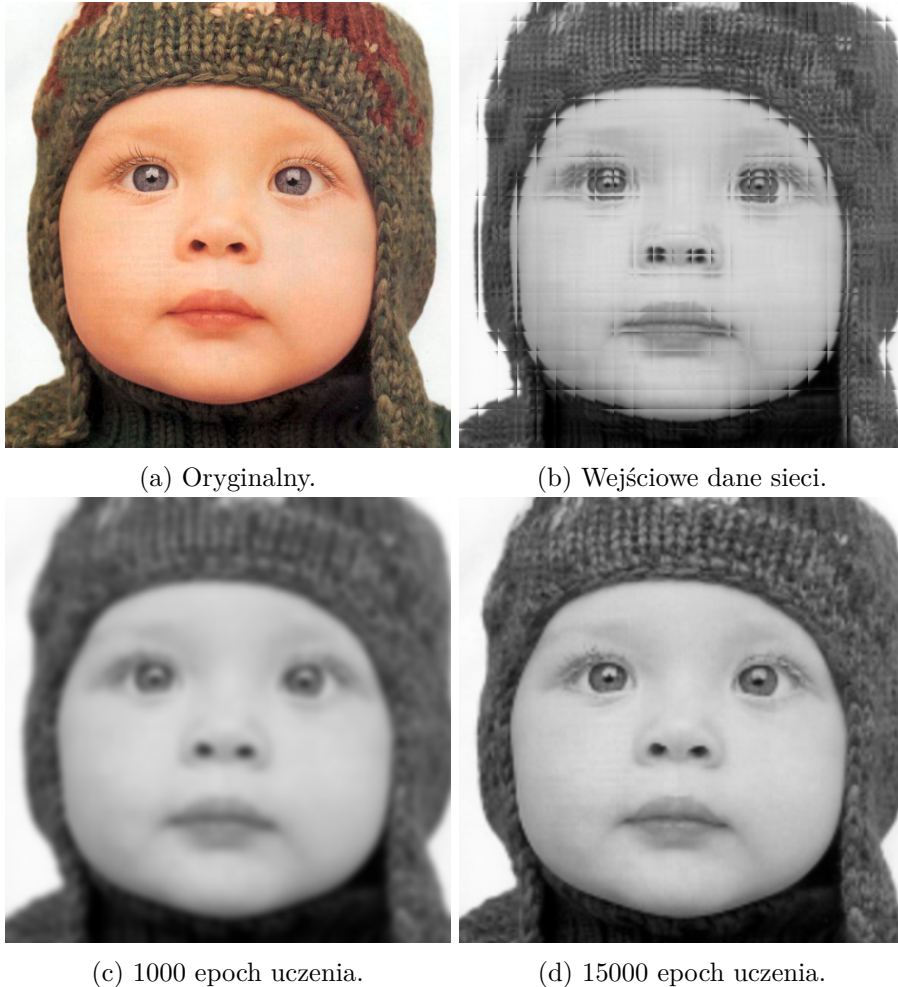
$$F_3(Y) = W_3 * F_2(Y) + B_3$$



Rysunek 2: Architektura SRCNN.

4.1.1 Wyniki działania algorytmu

Obejrzyć implementację która została użyta do uzyskania poniższych wyników można pod adresem: <https://github.com/tegg89/SRCNN-Tensorflow>. Artyków naukowy jest dostępny pod adresem <https://arxiv.org/abs/1501.00092>.



Rysunek 3: Wyniki działania algorytmu SRCNN.

4.2 EDSR

4.2.1 Wyniki działania algorytmu