Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

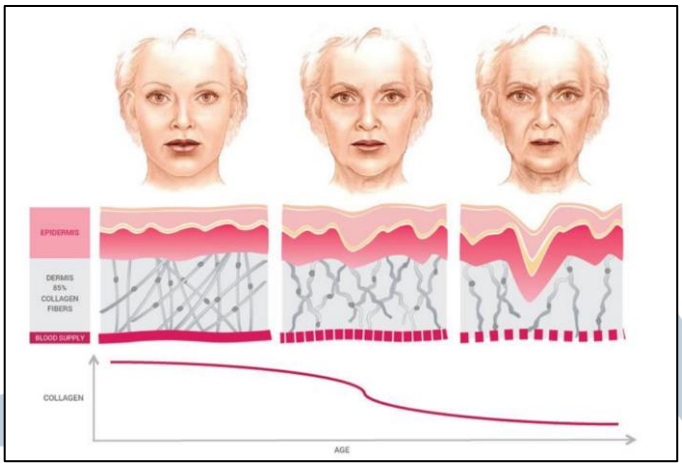
Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB II
LANDASAN TEORI

2.1 Keriput

Keriput adalah lipatan, garis atau kerutan yang muncul pada bagian wajah dan leher, umumnya kemunculan keriput pada wajah merupakan salah satu tanda penuaan, namun tidak jarang keriput muncul pada usia yang relatif muda atau sering disebut sebagai penuaan dini (Abriyani, 2018). Keriput terjadi karena berkurangnya kolagen pada kulit, selain itu munculnya keriput terjadi akibat adanya perubahan-perubahan faktor yang menunjang kulit di antaranya adalah:

a. Serabut kolagen dan serabut elastin yang berfungsi untuk menjaga kelenturan kulit berubah menjadi kaku dan tidak lentur, sehingga kehilangan daya elastisitasnya (Gambar 2.1).



Gambar 2.1 Ilustrasi Kulit Wajah Kekurangan Kolagen (Epilightnewskin, 2019)

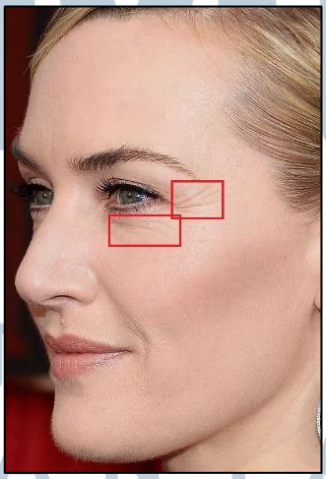
b. Tulang dan otot mengalami atrofi, jaringan lemak *subkutan* berkurang disertai lapisan kulit yang tipis, sehingga menyokong terbentuknya kerutan-kerutan dan lipatan-lipatan atau garis halus kulit yang nyata.

Implementasi arsitektur U-NET..., Shendy Harlim, FTI UMN, 2019

c. Pengaruh kontraksi otot-otot mimik yang tidak diikuti oleh kontraksi kulit yang sesuai, sehingga mengakibatkan garis-garis halus keriput muncul terutama di sekitar mulut, mata dan dahi (Abriyani, 2018).

Keriput tidak hanya datang akibat bertambahnya usia saja, tetapi beberapa faktor lain juga mempengaruhi kondisi kulit seseorang. Hal ini disebabkan oleh tingginya *stress*, tingkat polusi, perokok, pencemaran lingkungan, pancaran sinar matahari, radikal bebas, dan sinar UV (Susanti, 2013). Dalam keadaan tersebut apabila kulit tidak dirawat dengan baik maka kulit akan bertambah kering dan tampak lebih tua. kulit yang mulus dan halus akan menjadi kusam, keriput atau kurang elastis dan garis kerut-kerut kulit ini menjadi hal yang menakutkan, karena dianggap tidak menariknya penampilan (Susanti, 2013).

Pada umumnya ada dua jenis keriput yang muncul pada wajah yaitu keriput kasar dan keriput halus. Untuk keriput kasar biasanya muncul pada daerah garis-garis ekspresi pada wajah, seperti sudut luar mata, dahi, dan garis vertikal di kedua sisi mulut. Sedangkan untuk keriput halus berupa lekukan atau garis dangkal yang biasanya muncul pada daerah pergerakan muka seperti mata, mulut dan atas bibir (Abriyani, 2018). Gambar 2.2 menunjukkan area keriput pada wajah.



Gambar 2.2 Contoh Area Keriput pada Wajah

Implementasi arsitektur U-NET..., Shendy Harlim, FTI UMN, 2019

2.2 Segmentasi Citra

Segmentasi citra (*image segmentation*) adalah salah satu langkah penting dalam analisis sebuah citra (Awaluddin dan Darmo, 2010). Segmentasi citra secara otomatis dengan komputer adalah salah satu hal yang paling sulit dilakukan dalam pengolahan citra digital (Awaluddin dan Darmo, 2010). Segmentasi citra adalah yang paling banyak dipelajari dalam visi komputer (*computer vision*) (Rachmawati, dkk., 2016).

Segmentasi gambar semantik (*semantic image segmentation*) atau sering disebut klasifikasi tingkat piksel, adalah tugas pengelompokan bagian-bagian gambar yang termasuk di dalam kelas objek yang sama (Thoma, 2016). Pada Gambar 2.3 adalah contoh hasil dari segmentasi semantik. *Non-semantic segmenation* hanya mengelompokkan piksel berdasarkan karakteristik umum dari satu objek. Oleh karena itu tugas *non-semantic segmentation* tidak didefinisikan dengan baik, karena banyak segmentasi yang berbeda yang dapat diterima (Thoma, 2016).

Gambar 2.3 Contoh Segmentasi Semantik (Turgutlu, 2018)

NUSANTARA

2.3 Pembelajaran Mesin

Pembelajaran mesin (*machine learning*) adalah bidang ilmu yang sedang tumbuh pada bidang kecerdasan buatan atau *artificial intelligence* (AI), sangat penting karena memungkinkan mesin untuk mendapatkan kecerdasan seperti manusia tanpa pemrograman eksplisit (Das, dkk., 2015). Program AI melakukan hal-hal yang lebih menarik seperti pencarian web atau penandaan foto atau email anti-spam. Pembelajaran mesin dikembangkan sebagai kemampuan baru untuk komputer dan saat ini menyentuh banyak segmen industri dan *basic science* seperti *autonomous robotics* dan *computational biology* (Das, dkk., 2015).

Salah satu tipe pembelajaran mesin adalah pembelajaran terawasi (*supervised learning*) yaitu proses pembelajaran ini didasarkan pada perbandingan keluaran yang dihitung (*computed output*) dan keluaran yang diharapkan (*expected output*), pembelajaran ini mengacu pada menghitung *error* dan menyesuaikan *error* untuk mendapatkan keluaran yang diharapkan (Das, dkk., 2015). Salah satu metode yang dapat digunakan adalah jaringan syaraf tiruan.

2.4 Jaringan Syaraf Tiruan

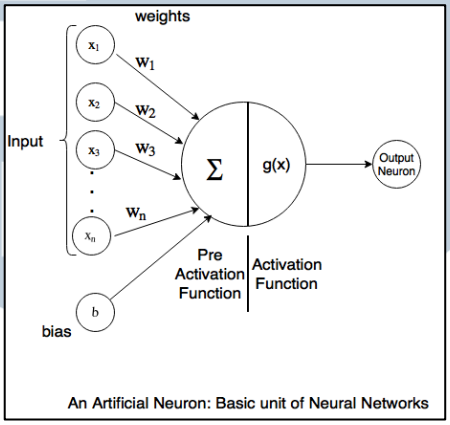
Jaringan syaraf tiruan atau *artificial neural network* (ANN) adalah imitasi dari otak manusia yang memiliki kemampuan untuk belajar hal baru, adaptasi terhadap lingkungan yang baru maupun yang berubah (Kukreja, dkk., 2016). ANN terdiri dari sejumlah besar elemen pemrosesan sederhana yang saling berhubungan satu sama lain dan juga berlapis. Mirip dengan neuron biologis ANN juga memiliki neuron yang dapat menerima masukan dari elemen lain yang kemudian masukan

Implementasi arsitektur U-NET..., Shendy Harlim, FTI UMN, 2019

akan ditimbang dan ditambahkan, hasilnya kemudian ditransformasikan menjadi keluaran (Sharma, dkk., 2012).

2.4.1 Neuron (Artificial Neuron)

ANN terdiri dari unit pemrosesan yang disebut neuron (Kukreja, dkk., 2016). Neuron merupakan sebuah sel sama seperti halnya dengan neuron pada syaraf manusia (Putri, 2014). Neuron terdiri dari masukan (dendrit), dan satu keluaran (sinapsis melalui akson) (Kukreja, dkk., 2016). Neuron memiliki fungsi yang menentukan aktivasi neuron (Kukreja, dkk., 2016).



Gambar 2.4 Model *Artificial Neuron* (Kukreja, dkk., 2016)

Pada Gambar 2.4, $x_1 \ldots x_n$ adalah masukan ke neuron. Bias juga ditambahkan ke neuron bersama dengan masukan. Nilai bias diinisialisasi dengan nilai 1. W_0 ... W_n adalah bobot. yang terkoneksi ke sinyal. Neuron menerima banyak masukan dari sumber yang berbeda, dan memiliki keluaran tunggal. Ada berbagai fungsi yang digunakan untuk aktivasi salah satunya adalah fungsi aktivasi yang paling umum digunakan adalah fungsi sigmoid (Kukreja, dkk., 2016).

Setiap input $x_1 \ldots x_n$ akan dikalikan dengan bobot yang dimiliki oleh satu neuron. Sebelum masuk fungsi aktivasi, semua input pada satu neuron akan dijumlahkan. Fungsi aktivasi yang digunakan adalah fungsi sigmoid, yang

Implementasi arsitektur U-NET..., Shendy Harlim, FTI UMN, 2019

merupakan perkiraan terus menerus dan terdiferensiasi dari fungsi langkah. Interkoneksi neuron individu tersebut membentuk jaringan saraf (Kukreja, dkk., 2016).

2.4.2 Layar (Layer)

Pada ANN, layar dapat berjumlah 2 (layar masukan dan layar keluaran) atau lebih (Putri, 2014). Setiap layar dapat terdiri dari beberapa neuron, sesuai yang diinginkan (Putri, 2014). Layar-layar penyusun ANN dapat dibagi menjadi tiga, yaitu sebagai berikut:

1. Layar masukan (*input layer*)

   Unit-unit di dalam layar masukan disebut unit-unit masukan. Unit-unit tersebut menerima pola masukan data dari luar yang menggambarkan suatu permasalahan yang akan dipecahkan.

2. Layar tersembunyi (*hidden layer*)

   Unit-unit di dalam layar tersembunyi disebut unit-unit tersembunyi di mana keluarannya tidak dapat secara langsung diamati. Layar tersembunyi merupakan layar dari elemen pemrosesan (neuron) yang tersembunyi dari koneksi langsung ke titik luar dari jaringan syaraf. Semua koneksi ke dan dari layar tersebut bersifat internal pada jaringan syaraf.

3. Layar keluaran (*output layer*)

   Unit-unit di dalam layar keluaran disebut unit-unit keluaran. Keluaran dari lapisan ini merupakan solusi jaringan syaraf terhadap suatu permasalahan (Putri, 2014).

Implementasi arsitektur U-NET..., Shendy Harlim, FTI UMN, 2019

2.4.3 Arsitektur Jaringan Syaraf Tiruan

Beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi antara lain:

1. Jaringan layar tunggal (*single layer network*)

   Dalam jaringan ini, sekumpulan neuron masukan dihubungkan langsung dengan sekumpulan neuron keluaran.

2. Jaringan layar jamak (*multi layer network*)

   Multi layar network merupakan perluasan dari layar tunggal. Dalam jaringan ini, selain unit masukan dan keluaran, ada unit lain yang sering disebut layar tersembunyi. Dimungkinkan pula ada beberapa layar tersembunyi. Sama seperti pada unit masukan dan keluaran, unit-unit dalam satu layar tidak saling berhubungan.

3. Jaringan *recurrent*

   Model jaringan *recurrent* mirip dengan jaringan layar tunggal ataupun ganda. Hanya saja, ada neuron keluaran yang memberikan sinyal pada unit masukan (sering disebut *feedback loop*) (Putri, 2014).

2.5 Fully Convolutional Network

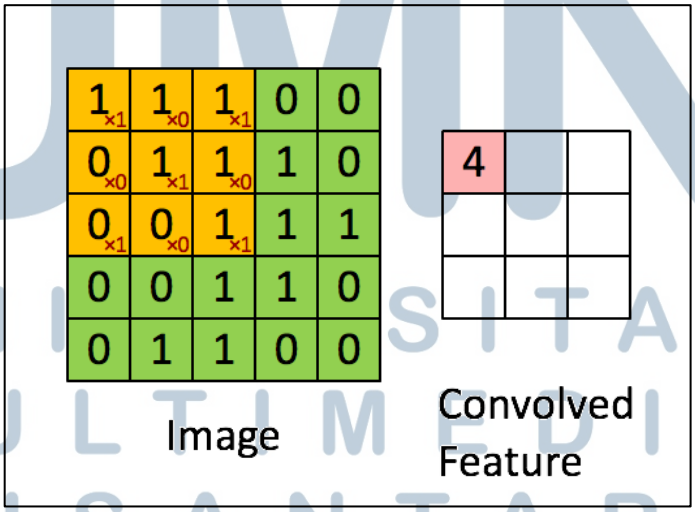
Setiap layar keluaran di dalam *convolutional network* (convnet) memiliki array tiga dimensi ukuran yaitu $h \times \mathrm{w} \times d$, di mana h dan w adalah dimensi spasial (*spatial dimension*), dan d adalah *feature* atau *channel dimension*. Layar pertama adalah gambar, dengan ukuran piksel $h \times \mathrm{w}$, dan d adalah *channels*. Lokasi pada layar tertinggi sesuai dengan lokasi di gambar yang terhubung dengan jalurnya, yang disebut sebagai bidang reseptif (*receptive fields*) (Shelhamer, dkk., 2016).

Implementasi arsitektur U-NET..., Shendy Harlim, FTI UMN, 2019

Convnet dibangun berdasarkan *translation invariant*. Komponen dasar (konvolusi, *pooling*, dan fungsi aktivasi) beroperasi pada *local input regions*, dan hanya bergantung pada koordinat spasial relatif. Ketika *general net* menghitung fungsi non-linear, jaringan yang terdiri hanya dengan layar dari bentuk ini menghitung filter non-linear, yang kita sebut *deep filter* atau *fully convolutional network* (FCN). FCN secara alami beroperasi pada masukan dengan ukuran berapa pun, dan menghasilkan keluaran yang sesuai dimensi spasial (Shelhamer, dkk., 2016).

2.5.1 Konvolusi

Konvolusi (*convolution*) adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada keluaran fungsi lain secara berulang. Dalam pengolahan citra, konvolusi berati mengaplikasikan sebuah *kernel* (kotak kuning) pada citra disemua *offset* yang memungkinkan seperti yang ditunjukkan pada Gambar 2.5. Kotak hijau secara keseluruhan adalah citra yang akan dikonvolusi. *Kernel* bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada gambar di sebelah kanannya (Wijaya, dkk., 2016).

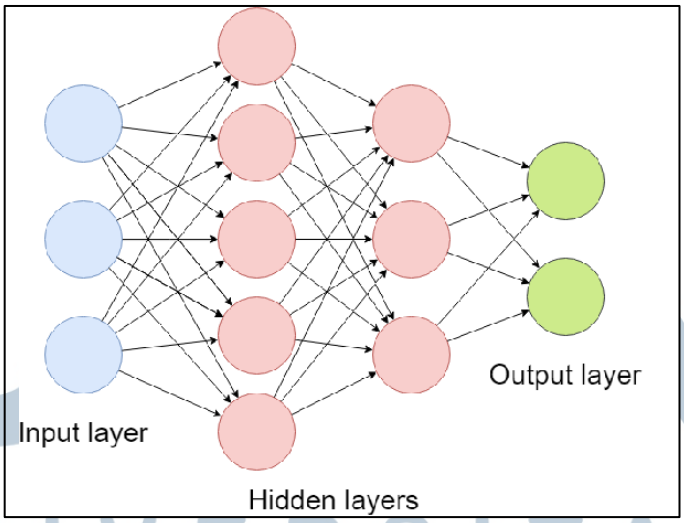


Gambar 2.5 Operasi Konvolusi (Wijaya, dkk., 2016)

Implementasi arsitektur U-NET..., Shendy Harlim, FTI UMN, 2019

Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra masukan. Konvolusi akan menghasilkan transformasi linear dari data masukan sesuai informasi spasial pada data. Bobot pada layar tersebut menspesifikasikan *kernel* konvolusi yang digunakan (Wijaya, dkk., 2016).

Setiap angka dalam matriks *kernel* atau filter adalah bobot yang harus dipelajari selama waktu pelatihan. Setiap filter memiliki jumlah saluran yang sama dengan masukan langsungnya. Misalnya, memasukkan gambar dengan tiga saluran bentuk 256x256x3 piksel dapat memiliki filter geser 5x5x3 dan menghitung produk titik di atasnya. Filter dalam lapisan tersembunyi memiliki saluran sebanyak layar sebelum mereka memiliki filter. Di setiap lokasi spasial, produk titik untuk setiap saluran masukan dihitung dan dijumlahkan bersama (Salumaa, 2018). Gambar 2.6 merupakan contoh tiga layar *neural network* dengan dua layar tersembunyi dan satu layar keluaran (Salumaa, 2018).

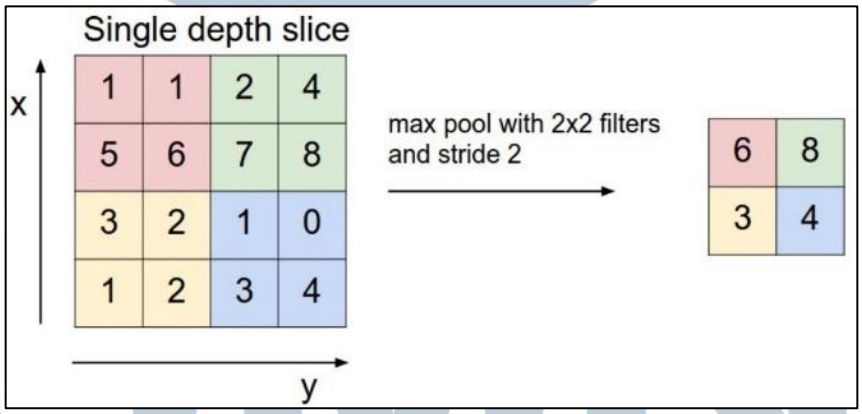


Gambar 2.6 Jaringan *neural network* (Salumaa, 2018)

UNIVERSITAS MULTIMEDIA NUSANTARA

2.5.2 Subsampling

Subsampling adalah proses mereduksi ukuran sebuah data citra. Dalam pengolahan citra, *subsampling* juga bertujuan untuk meningkatkan invarian posisi dari fitur. Dalam sebagian besar *convolutional neural network* (CNN), metode *subsampling* yang digunakan adalah *max pooling*. *Max pooling* membagi *output* dari layar konvolusi menjadi beberapa *grid* kecil lalu mengambil nilai maksimal dari setiap *grid* untuk menyusun matriks citra yang telah direduksi seperti yang ditunjukkan pada Gambar 2.7. *Grid* yang berwarna merah, hijau, kuning dan biru merupakan kelompok *grid* yang akan dipilih nilai maksimumnya. Sehingga hasil dari proses tersebut dapat dilihat pada kumpulan *grid* di sebelah kanannya. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun objek citra mengalami translasi (pergeseran) (Wijaya, dkk., 2016).



Gambar 2.7 Operasi *Max Pooling* (Wijaya, dkk., 2016)

2.5.3 Transposed Convolution

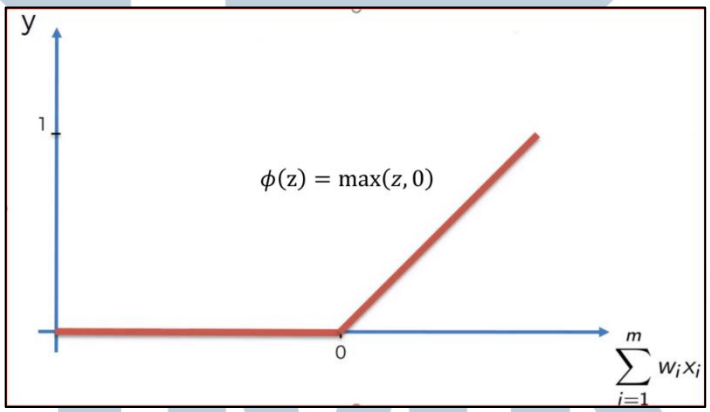
Transposed convolution juga disebut *fractionally strided convolutions* atau *deconvolutions*. *Transposed convolution* muncul dikarenakan kebutuhan untuk menggunakan transformasi yang berlawanan arah dari normal konvolusi yaitu dari hasil output dari konvolusi untuk dikembalikan ke bentuk masukan sebelum konvolusi (Dumoulin & Visin, 2018). Sebagai contoh, salah satu tujuan

digunakannya *transposed convolution* adalah memproyeksikan *feature maps* ke dimensi yang lebih besar (Zhang A. , dkk., 2019).

2.5.4 Fungsi Aktivasi

Fungsi aktivasi (*activation function*) adalah fungsi yang memungkinkan sebuah jaringan syaraf tiruan untuk dapat mentransformasi data masukan menjadi dimensi yang lebih tinggi sehingga dapat dilakukan pemotongan *hyperlane* sederhana yang memungkinkan dilakukan klasifikasi (Wijaya, dkk., 2016). Salah satu fungsi aktivasi pada keluaran dari lapisan konvolusi adalah *Rectified Linear Unit* (ReLU). Secara matematika dapat didefinisikan pada Rumus 2.1 (Salumaa, 2018).

$$y = max(0, x) \quad \dots (2.1)$$

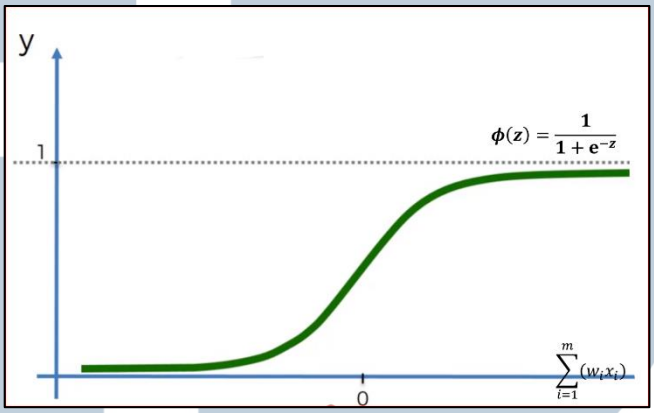


Gambar 2.8 Fungsi ReLU (Khandelwal, 2018)

Pada Rumus 2.1, x mewakili matriks keluaran setelah konvolusi (Salumaa, 2018).Setelah operasi ReLU dilakukan, maka semua angka negatif dalam matriks *feature map* akan ditetapkan nol dan nilai lainnya tidak berubah (Salumaa, 2018). ReLU adalah fungsi aktivasi yang paling umum digunakan dalam jaringan syaraf tiruan (Salumaa, 2018). Secara visual dapat dilihat pada Gambar 2.8.

$$sigmoid(x) = \frac{1}{1 + exp(-x)} \quad \dots (2.2)$$

Pada Rumus 2.2, fungsi sigmoid akan mentranformasikan nilai ke interval antara 0 dan 1 (Zhang, dkk., 2019). Fungsi sigmoid umumnya digunakan dalam jaringan saraf awal (Zhang, dkk., 2019). Turunan dari fungsi sigmoid diplot. Di bawah. Ketika masukan mendekati 0, fungsi sigmoid mendekati transformasi linear (Zhang, dkk., 2019). Secara visual dapat dilihat pada Gambar 2.9.

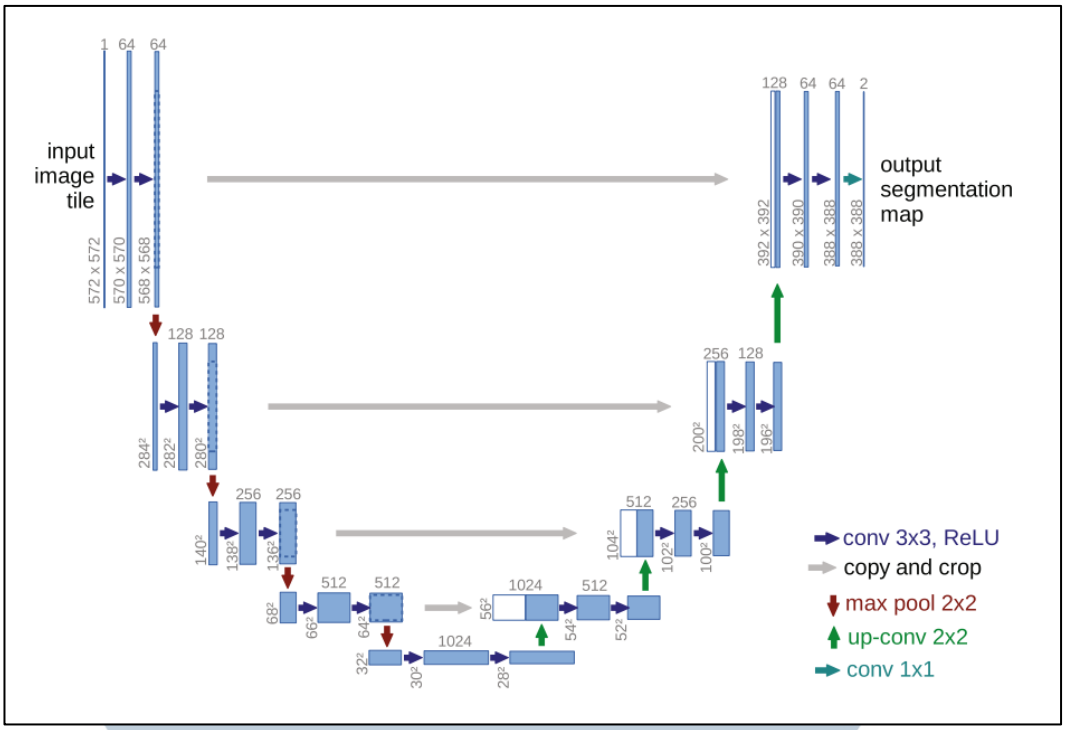


Gambar 2.9 Fungsi Sigmoid (Khandelwal, 2018)

2.6 U-Net

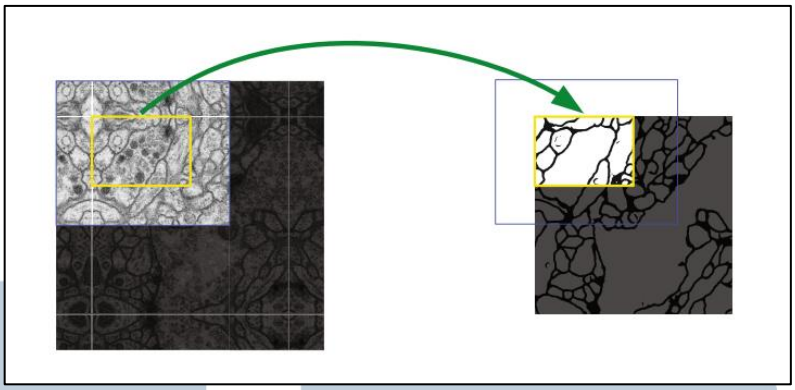
U-Net adalah pengembangan dari arsitektur yang elegan yaitu "*Fully Convolutional Network*" yang dimodifikasi dan dikembangkan sehingga arsitektur ini dapat berfungsi dengan sangat sedikit gambar *training* dan menghasilkan segmentasi yang lebih tepat (*precise*). Gagasan utamanya adalah untuk melengkapi *contracting network* yang biasa dengan *layer* yang berturut-turut, di mana operator *pooling* diganti dengan operator *upsampling*. Oleh karena itu, layar ini meningkatkan resolusi keluaran. Untuk lokasi, fitur resolusi tinggi dari *contracting path* dikombinasikan dengan keluaran *upsampled*. Layar konvolusi berturut-turut kemudian dapat belajar untuk mengumpulkan keluaran yang lebih tepat (*precise*) berdasarkan informasi ini. Secara visual dapat dilihat pada Gambar 2.10 (Ronneberger, dkk., 2015).

Implementasi arsitektur U-NET..., Shendy Harlim, FTI UMN, 2019



Gambar 2.10 Arsitektur U-Net (Ronneberger, dkk., 2015)

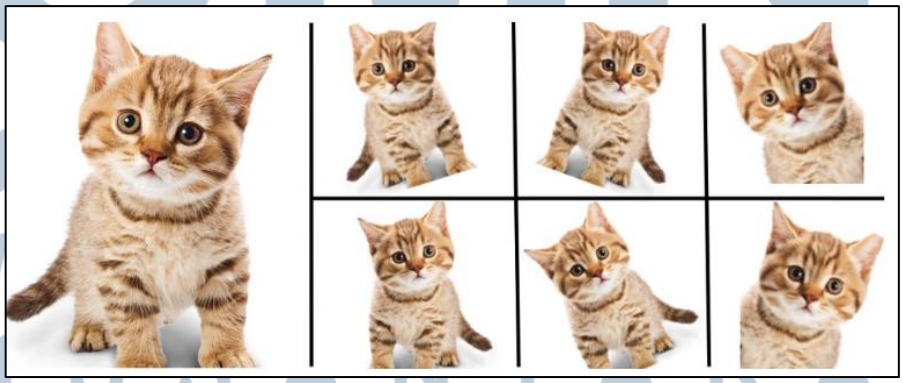
Modifikasi arsitektur yang penting lainnya adalah pada bagian *upsampling*, yaitu dengan memiliki jumlah *feature channels* yang besar, sehingga memperbolehkan jaringan untuk menyebarkan informasi kontek (*context information*) ke layar resolusi yang lebih tinggi. Akibatnya, *expansive path* lebih atau kurang simetris dengan *contracting path* dan menghasilkan arsitektur berbentuk huruf U. Jaringan ini tidak memiliki *fully connected layers* dan menggunakan bagian yang valid dari setiap konvolusi, yaitu hanya *segmentation map* berisi piksel, yang konteks penuh tersedia pada gambar masukan. Strategi ini memungkinkan segmentasi *arbitrarily large images* dengan strategi *overlap-tile* dapat dilihat pada Gambar 2.11. Untuk memprediksi piksel di *border region* pada gambar, *missing context* diekstrapolasi dengan *mirroring the input image*. Strategi *tiling* ini penting untuk menerapkan jaringan ke gambar berukuran besar, karena jika tidak resolusi gambar akan dibatasi oleh memori GPU (Ronneberger, dkk., 2015).



Gambar 2.11 *Overlap-tile Strategy for Seamless Segmentation* (Ronneberger, dkk., 2015)

2.7 Augmentasi Data

Augmentasi data sangat penting untuk mengajarkan jaringan sifat invarian dan *robustness properties* yang diinginkan, ketika hanya sedikit sampel *training* yang tersedia (Ronneberger, dkk., 2015). Augmentasi data merupakan salah satu cara yang dapat digunakan untuk mengurangi *overfitting* pada model, yaitu dengan menambah jumlah data *training* menggunakan informasi pada data *training* yang sudah ada, sehingga dapat meningkatkan akurasi dari permasalahan klasifikasi (Wang dan Perez, 2017). Terdapat beberapa cara yang dapat digunakan untuk melakukan augmentasi yaitu *flips*, *gaussian noise*, *jittering*, *rotations*, *shears*, dan lain-lain (Hussain, dkk., 2017). Contoh augmentasi data dapat dilihat pada Gambar 2.12.



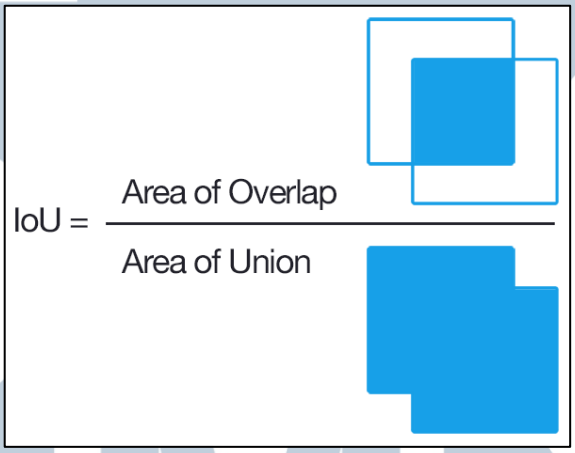
Gambar 2.12 Contoh Augmentasi Data (Raj, 2018)

Implementasi arsitektur U-NET..., Shendy Harlim, FTI UMN, 2019

2.8 Intersection over Union

Intersection over Union (IoU) mengukur seberapa baik segmentasi yang diprediksi dengan anotasi *ground truth* dengan membagi persimpangan dua segmen dengan gabungan antara anotasi *ground truth* dan hasil prediksi (Falk, dkk., 2019). Misalkan $S := \{s_1, \dots, s_n\}$ adalah satu set N piksel. Misalkan $G \subset S$ adalah himpunan piksel yang dimiliki oleh segmen *ground truth* dan $P \subset S$ adalah himpunan piksel yang dimiliki oleh segmen prediksi. IoU dapat didefinisikan pada Rumus 2.3 (Falk, dkk., 2019). Secara visual dapat dilihat pada Gambar 2.13.

$$IoU = \frac{|G \cap P|}{|G \cup P|} \quad \dots (2.3)$$



Gambar 2.13 Persamaan *Intersection over Union* (Rosebrock, 2016)

Implementasi arsitektur U-NET..., Shendy Harlim, FTI UMN, 2019