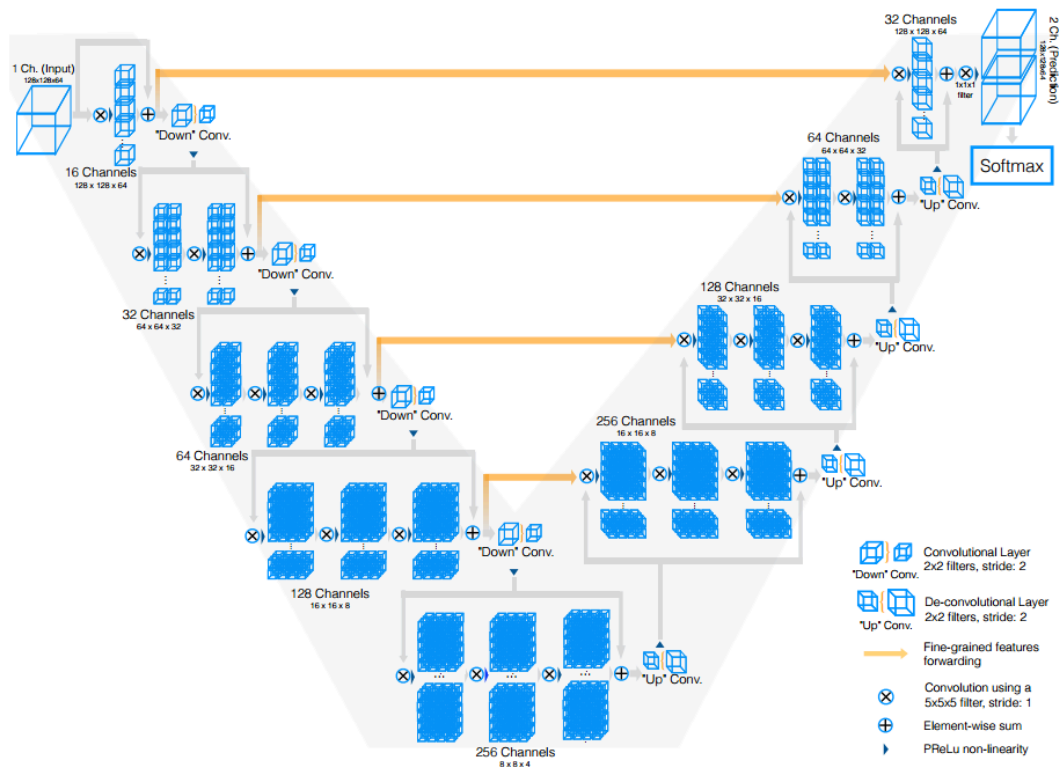


Arsitektur 2D V-Net



Gambar 3.2. Arsitektur V-Net [50]

Pada gambar arsitektur 2D V-Net, jumlah channels pada encoder selalu bertambah dari 16, 32, 64, 128. Hal ini dilakukan untuk meningkatkan kapasitas model untuk mempelajari informasi yang lebih kompleks. Dengan meningkatkan jumlah channels, model dapat belajar untuk mengenali fitur yang lebih halus dan lebih spesifik. Pada tingkat konvolusi pertama, model belajar untuk mengenali fitur lokal sederhana, seperti tepi dan sudut. Pada tingkat konvolusi kedua, model belajar untuk mengenali fitur yang lebih kompleks, seperti bentuk dan pola. Pada tingkat konvolusi ketiga, model belajar untuk mengenali fitur yang paling kompleks, seperti tekstur dan struktur. Dengan meningkatkan jumlah channels pada encoder, model dapat belajar untuk mengenali fitur-fitur ini dengan lebih akurat. Hal ini dapat meningkatkan kinerja model pada tugas segmentasi citra volumetrik, seperti segmentasi organ, tumor, dan jaringan.

3.2.1. Proses *Encoder*

Pada tingkat konvolusi pertama pada *encoder*, model belajar untuk mengenali fitur lokal sederhana, seperti tepi dan sudut. Jumlah *channels* yang kecil, seperti 16, memungkinkan model untuk mempelajari fitur-fitur ini dengan cepat dan efisien. Setelah itu,

model menggunakan lapisan *pooling* untuk mengurangi ukuran citra input menjadi setengah dari ukuran aslinya. Hal ini dilakukan untuk mengurangi jumlah informasi yang perlu diproses oleh model.

Pada tingkat konvolusi kedua, model belajar untuk mengenali fitur yang lebih kompleks, seperti bentuk dan pola. Jumlah *channels* yang bertambah dari 16 menjadi 32 memungkinkan model untuk mempelajari fitur-fitur ini dengan lebih detail. Setelah itu, model menggunakan lapisan *pooling* untuk mengurangi ukuran citra input menjadi seperempat dari ukuran aslinya.

Pada tingkat konvolusi ketiga, model belajar untuk mengenali fitur yang paling kompleks, seperti tekstur dan struktur. Jumlah *channels* yang bertambah dari 32 menjadi 64 memungkinkan model untuk mempelajari fitur-fitur ini dengan lebih akurat. Setelah itu, model menggunakan lapisan *pooling* untuk mengurangi ukuran citra input menjadi seperdelapan dari ukuran aslinya.

Pada tingkat konvolusi keempat, jumlah *channels* bertambah dari 64 menjadi 128. Jumlah *channels* yang tinggi memungkinkan model untuk mempelajari fitur-fitur global dari citra input. Setelah itu, model menggunakan lapisan *pooling* untuk mengurangi ukuran citra input menjadi sepersepuluh dari ukuran aslinya.

3.2.2. Proses Pada *Bridge*

Setelah itu pada *bridge* terdapat 256 *channels*. Hal ini dilakukan untuk menghubungkan *encoder* dan *decoder*. *Encoder* mengekstrak fitur dari citra input, sedangkan *decoder* membangun kembali citra *output*. *Bridge* menghubungkan kedua bagian ini dengan cara menggabungkan fitur dari *encoder* dengan fitur dari *decoder*.

Pada tingkat koneksi pertama, model menggabungkan fitur dari tingkat konvolusi ketiga pada *encoder* dengan fitur dari tingkat dekonvolusi pertama pada *decoder*. Pada tingkat koneksi kedua, model menggabungkan fitur dari tingkat konvolusi kedua pada *encoder* dengan fitur dari tingkat dekonvolusi kedua pada *decoder*.

3.2.3. Proses *Decoder*

Pada *decoder*, model menggunakan proses *upsampling* untuk membangun kembali citra *output* dari fitur-fitur yang diekstrak oleh *encoder*. *Upsampling* dilakukan dengan menggunakan lapisan konvolusi *transpose*. Lapisan konvolusi *transpose* adalah lapisan konvolusi yang menggunakan filter berukuran kecil untuk meningkatkan ukuran citra input. Pada citra arsitektur 2D V-Net, *upsampling* dilakukan dengan menggunakan lapisan

dekonvolusi. Lapisan dekonvolusi adalah lapisan konvolusi *transpose* yang khusus dirancang untuk digunakan dalam arsitektur 2D V-Net.

Pada tingkat dekonvolusi pertama, jumlah *channels* berkurang dari 256 menjadi 128. Hal ini dilakukan untuk mengurangi jumlah informasi yang perlu diproses oleh model. Setelah itu, model menggunakan lapisan dekonvolusi untuk meningkatkan ukuran citra input menjadi dua kali dari ukuran aslinya.

Pada tingkat dekonvolusi kedua, jumlah *channels* berkurang dari 128 menjadi 64. Setelah itu, model menggunakan lapisan dekonvolusi untuk meningkatkan ukuran citra input menjadi empat kali dari ukuran aslinya.

Pada tingkat dekonvolusi ketiga, jumlah *channels* berkurang dari 64 menjadi 32. Setelah itu, model menggunakan lapisan dekonvolusi untuk meningkatkan ukuran citra input menjadi delapan kali dari ukuran aslinya.

Pada tingkat dekonvolusi terakhir, jumlah *channels* berkurang dari 32 menjadi 1. Lapisan dekonvolusi terakhir ini menghasilkan citra *output* dengan ukuran yang sama dengan citra *input*. Setelah itu, model menggunakan lapisan *softmax* untuk menghasilkan *output* yang memiliki nilai antara 0 dan 1. Nilai 0 menunjukkan bahwa pixel tersebut tidak termasuk ke dalam target, sedangkan nilai 1 menunjukkan bahwa pixel tersebut termasuk ke dalam target.