2. VGGNET (2014년) - Very Deep Convolutional Networks for Large-Scale Image Recognition

VGGNet

- 연구내용
  - 연구팀이 대규모 이미지 인식에 있어서 컨볼루션 네트워크의 깊이가 정확오데 어떤 영향을 미치는지 조사함
  - VGGNet연구팀은 3 x 3 Conv filter를 여러개 쌓아 기존 CNN 모델의 layer 개수를 deep하게 늘렸음

- VGG Net 구조
  - 훈련에서는 ConvNet에 224 X 224로 고정된 RGB 영상을 입력으로 받도록 함
  - 전처리 : 훈련 집합에 대해 RGB값의 평균을 각 픽셀에 빼주는 것
    - 데이터 정규화(normalization)의 한 형태 - 이 과정을 통해 모델이 학습하기에 더 용이한 데이터 형태로 만들어 주는게 목적
      - 데이터 중심화 : 각 채널별로 평균값을 빼주면 데이터의 중심을 0 주변으로 옮김
        - 데이터 포인트들이 평균값을 중심으로 분포하게 만들어 학습과정에서 가중치의 업데이트가 더 안정적이고 효율적으로 이루어지도록 도움
      - 학습 과정의 가속화 : 데이터를 정규화함으로써 그래디언트 기반 최적화 알고리즘이 더 빠르고 안정적으로 수렴하게 할 수 있음
        - 데이터의 분산이 감소하므로 파라미터 업데이트 시 발생할 수 있는 급격한 변동을 줄여줌
      - 일반화 성능 향상 : 평균값 정규화는 모델이 특정 색상의 밝기나 대비에 과도하게 의존하는 것을 방지하여, 다양한 환경에서 촬영된 이미지에 대한 모

델의 일반화 능력을 향상 시킴

- 모델이 색상의 절대적인 값보다는 객체의 형태나 패턴과 같은 구조적인 특징을 학습하는데 더 집중하게 됨

- 이러한 전처리 과정은 특히 깊은 신경망에서 중요하며, 네트워크의 깊이가 깊어질수록 파라미터 업데이트 과정에서 발생할 수 있는 문제를 완화시켜 주고 모델의 학습 속도와 성능을 개선하는데 도움을 줌

- Conv Layer
  - 3 × 3 Conv filter를 사용
    - 3 × 3 사이즈가 이미지 요소의 left, right, up, down 등을 파악할 수 있는 최소한의 receptive field이기 때문에 3 x 3 사이즈를 사용
      - receptive field : 신경망의 특정 출력이 입력 이미지의 어느 부분에 의해 영향을 받는지를 나타내는 용어 즉, 신경망에서 한 뉴런이 '보는' 입력 데이터의 영역
        - 최소한의 커버리지 제공 : 3 x 3 사이즈는 한 번의 연산으로 주변 픽셀들에 대한 정보를 효과적으로 수집할 수 있는 최소한의 크기임, 1x1 필터는 주변 컨텍스트 없이 하나의 픽셀만을 고려하며, 5x5 이상의 크기는 더 많은 파라미터와 계산량을 요구함
          - 3×3필터는 주변의 상, 하, 좌, 우 뿐만 아니라 대각선 방향의 정보도 포함하여 이미지의 기본적인 구주와 패턴을 파악하는데 충분한 컨텍스트를 제공함
        - 효율성과 효과성의 균형 : 3 x 3 컨볼루션 필터는 파라미터의 수와 계산량 측면에서 효율적임, 이 크기의 필터를 여러 층에 걸쳐 적용하면, 더 깊은 층으로 갈수록 확장되는 receptive field를 통해 이미지의 더 넓은 영역을 커버할 수 있음
          - 예를 들어, 3개의 3 x 3 컨볼루션 층을 연속으로 사용하면 최종적으로 7 x 7크기의 영역에 대한 정보를 처리할 수 있게 되므로 깊이가 깊어질수록 더 복잡하고 추상적인 특징을 학습할 수 있음
          - 컨볼루션 신경망의 여러층을 거치면서 feacture map이 점점 작아지지만, 각 유닛의 receptive field가 더 넓은 영역을 '보게'됨
          - 첫 번째 컨볼루션 층에서 필터는 매우 구체적으로 작은 영역에 집중
            네트워크를 거쳐 더 깊은 층으로 갈수록 각 층의 피처 맵은 원

본 이미지의 더 넓은 영역에 대한 정보를 종합하게 됨
그 결과, 더 깊은 층에서는 원본 이미지의 더 큰 부분을 대표하는 추상적인 특징들을 포착할 수 있게되어 피처 맵의 크기는 점차 작아지지만, 각 피처 맵의 유닛이 처리하는 이미지 영역(receptive field)은 더 넓어

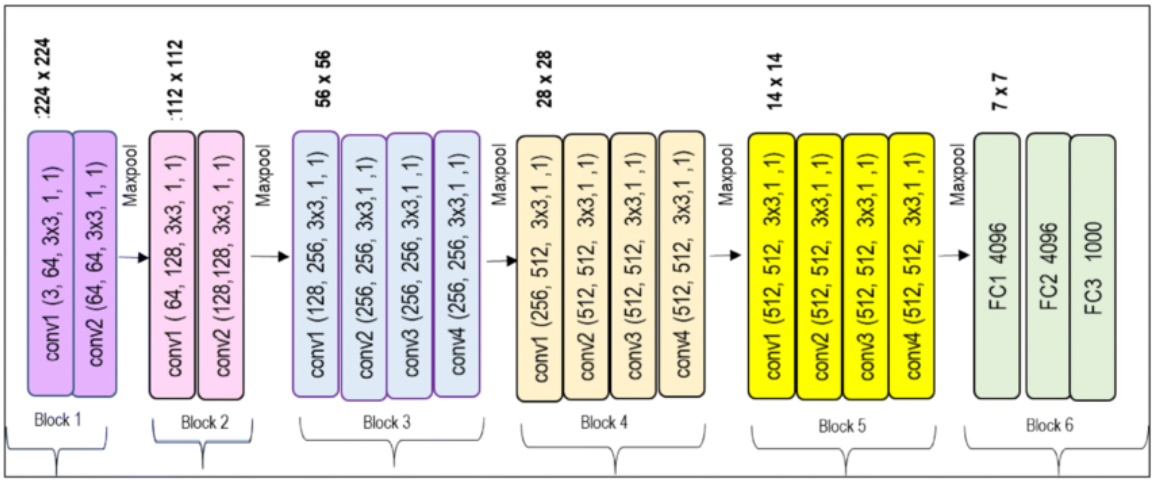
- 1 × 1 Conv filter도 사용하는데, 차원을 줄이고 non-linearity를 증가시키기 위함
  - 입력 채널의 선형변환(이후엔 비선형 변환)
    - 차원축소 : 1x1 컨볼루션은 피처 맵의 깊이(채널 수)를 줄일 수 있음
      이는 네트워크의 파라미터 수를 감소시키고, 계산 효율을 향상시키는데 도움이 됨
      - 예를 들어, 256개의 채널을 가진 피처 맵에 1x1 컨볼루션 필터를 64개 적용한다고 하면 결과적으로 64개의 채널을 가진 피처 맵을 얻을 수 있음
        이는 피처 맵의 깊이를 크게 줄이면서도 중요한 정보는 보존할 수 있음
    - 비선형성 증가 : 컨볼루션 신경망에서는 활성화 함수를 통해 비선형성을 도입
      비선형성은 모델이 더 복잡한 패턴과 데이터의 비선형 관계를 학습하는데 필수적
      1x1 컨볼루션 층 후에 활성화 함수(예 : ReLU)를 적용하면, 새로운 비선형 변환을 추가할 수 있음
      이렇게 함으로써 각 1x1컨볼루션은 각 픽셀의 값에 독립적으로 작용하여, 피처 맵의 각 위치에 비선형성을 도입함
- Stride , Padding = 1로 설정
  - convolution stride는 1픽셀로 고정 → conv 계층의 입력의 공간적 padding이 발생하더라도 convolution 이후에 공간적인 화질은 보존되도록함
    - Stride를 1로 설정하는 이유

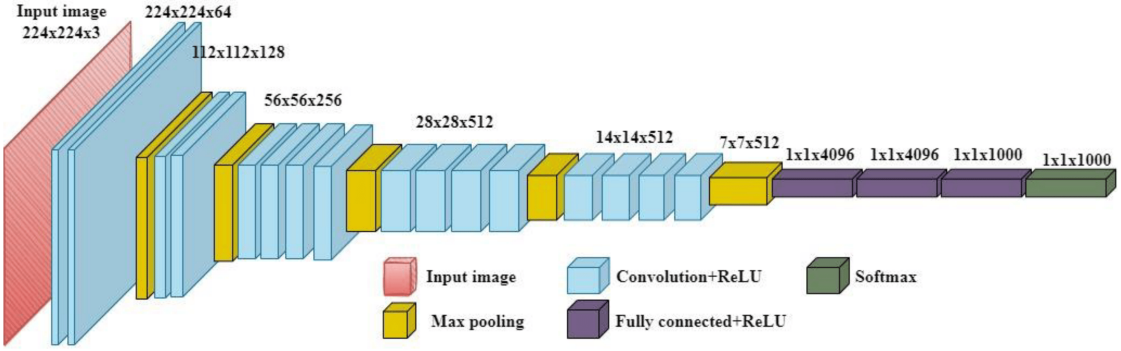
        - Stride 필터가 입력 데이터를 처리할 때 이동하는 거리를 나타냄
          Stride가 1이면 필터는 한 칸씩 움직이며, 이렇게 함으로써 입력 데이터의 모든 정보를 가장 정밀하게 처리할 수 있음
            - Stride를 1로 설정하면 피처 맵의 크기가 크게 줄지 않아 입력 데이터의 공간적 해상도를 보존할 수 있음
      - Padding을 1로 설정하는 이유
        - Padding은 입력 데이터의 가장 자리 주변에 추가되는 픽셀의 양을 나타냄
          3 x 3 컨볼루션 층에서 padding을 1로 설정하면, 입력 데이터의 각 변 주변에 한 픽셀의 가장자리가 추가 됨
          이렇게 하는 이유는 컨볼루션 연산 후에도 출력 피처 맵의 크기가 입력 데이터의 크기가 동일하게 유지되도록 하기 위함
            - 입력 데이터의 가장자리 정보가 손실되는 것을 방지하고 공간적 해상도를 보존하는 효과가 있음
  - Pooling Layer
    - Conv layer 다음에 적용되며, 총 5개의 max pooing layer로 구성
      - Pooling Layer의 역할 : 피처 맵의 공간적 크기를 줄이고 계산 부하를 감소
        - 피처 맵의 공간적 변화에 대한 강인성을 증가
      - Max Pooling Layer : Max Pooling은 각 피처 맵 영역에서 가장 큰 값을 선택하여 새로운, 축소된 피처 맵을 생성
        - 피처 맵 내에서 가장 두드러진 특징을 강조하는 효과
      - 5개의 Max Pooling Layer : 네트워크에 5개의 max pooling 계층이 있다는 것은, 신경망의 다양한 깊이에서 공간적 해상도를 점차적으로 줄이며 중요한 정보를 추출하고 있음을 의미
        - 각각의 max pooling 계층은 하나 이상의 컨볼루션 계층 뒤에 위치하여, 계층별로 추출된 특징들을 더욱 간결하게 요약하고 학습
      - 신경망이 효율적으로 특징을 학습하고 입력 이미지의 다양한 크기와 비율에 대해 강인하도록 만들어 줌
      - Max Pooling 계층을 통해 네트워크는 불필요한 정보를 줄이고, 중요한 정보를 유지하며 과적합을 방지하는데 도움을 받음
    - Max-pooling은 2X2 픽셀창에 stride가 2로 처리가 됨

- 2×2 픽셀 창 : 각 pooling 연산은 2x2의 크기를 가진 픽셀 창(window) 내에서 수행
    - 해당 창 내의 픽셀 중에서 가장 큰 값을 선택하여, 해당 창을 대표하는 값으로 사용
- Stride 2 : Stride는 필터(혹은 창)가 입력 피처 맵 위를 이동하는 거리를 나타냄
- Stride가 2라는것은 각 Pooling 연산 후 필터가 2 픽셀씩 건너뛰면서 이동한다는 것을 의미
    - 각 pooling 연산의 결과는 원본 피처 맵에서 겹치지 않는 영역을 기반으로 함

- FC Layer
    - 처음 두 FC Layer는 4,096 채널, 마지막 FC Layer는 1,000 채널
        - FC는 Fully Connected의 약자로 일반적으로 신경망의 마지막 부분에서 입력된 특징들을 토대로 최종 결론을 도출하는데 사용
        - 컨볼루션층과 풀링층을 통해 추출된 모든 특징을 모아 최종적인 출력을 생성하기 위한 층
            - FC 층은 학습된 특징들을 바탕으로 복잡한 함수를 모델링하여 이미지 분류, 회귀 등의 문제를 해결할 수 있음
        - 4,096 채널을가진 FC 층이 2개가 있는 이유
            - 추상화 레벨 증가 : 각각의 FC 층은 입력된 정보에 대한 또 다른 수준의 추상화를 제공
            - 첫 번째 FC 층은 이전 층에서 추출된 특징들을 기반으로 복잡한 패턴을 학습
            - 두 번째 FC 층은 첫 번째 FC 층의 출력을 사용하여 더 높은 수준의 추상 패턴을 학습
        - 비선형성 추가 : 여러 FC 층을 사용함으로써 모델에 더 많은 비선형성을 도입할 수 있음
            - 각 층 사이에 비선형 활성화 함수(ReLU)를 적용하면 모델이 더 복잡한 함수를 학습할 수 있게 됨
        - 과적합 방지 : 여러 층을 사용하면 각 층에서 학습할 파라미터의 수를 분산시킬 수 있음

        - 각 층이 보다 일반화된 특징을 학습하는 데 도움을 주어 과적합의 위험을 줄임
    - 학습용이성 : 깊은 네트워크는 특히 다양한 수준의 특징과 패턴을 분리해내는 데 유리하기 때문에 하나의 층보다는 여러층을 쌓는 것이 네트워크가 다양한 특징을 더 잘 학습하고, 이를통해 더 복잡한 문제를 해결할 수 있게 만듬
    - 두 개 이상의 FC 층을 사용하는 것은 깊은 신경망 설계에서 흔히 볼 수 있는 패턴
        - 네트워크가 더 복잡한 패턴을 인식하고, 더 나은 일반화를 달성할 수 있도록 함
    - 마지막 FC Layer
        - 마지막 FC Layer가 1,000 채널을 가진다는 것은 1,000개의 뉴런을 갖고 있다는 것을 의미함
        - 각 뉴런은 최종적으로 하나의 특정 클래스를 나타낼 수 있으며 신경망이 분류 문제를 해결할 때 이용
            - 예를 들어, ImageNet 분류 문제에서는 1,000개의 다른 카테고리가 있으므로 마지막 FC 층은 각 카테고리에 대한 점수를 계산하는데 사용 됨
            - 1,000개의 채널은 모두 다른 이미지이며 각 사진이 어떤 클래스를 나타내는지 점수를 매기는 용도
                - 예를 들어 분류작업이라면 가장 높은 확률 값을 가진 클래스가 '고양이'로 나온다면 출력은 '고양이'로 나타냄
- FC Layer의 마지막 부분에서만 Soft-max Layer를 적용해주고, 그 외 모든 layer에는 ReLU를 적용하여 비선형성을 변환을 추가함
- AlexNet에서 사용한 LRN 기법은 성능 개선은 없고 메모리 사용량 및 연산 시간만 늘어났기에 사용하지 않음





- **CONFIGURATIONS**
  - Depth에 따라 모델 구조가 조금씩 변형 되었으며, 11 Depth인 A구조에서부터 19 Depth인 E구조까지 있음
  - Conv Layer의 폭은 64에서부터 시작해 max pooling layer를 통과할 때 마다 2의 제곱만큼 커져, 최대 512까지 커짐
  - Depth가 늘어남에도 더 큰 Conv Layer를 사용한 얕은 신경망보다 오히려 파라미터 수가 줄어들었다고 설명

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144