

# Main acronyms

# 1. TSA: Textual Sentiment Analysis

## 1. 텍스트 감정 분석

예) 누군가가 "이 영화 진짜 감동이었어!"라고 썼다면 TSA는 "긍정"이라고 판단해요. 반대로 "완전 지루하고 시간 낭비였다"는 문장은 "부정"으로 분류됩니다. 이 기술은 리뷰 분석, 여론 조사, 고객 피드백 이해 등에 널리 쓰여요.

## 2. FER: Facial Expression Recognition

1. 얼굴 표정 인식
2. 사람 얼굴을 보고 그 사람이 어떤 감정을 느끼고 있는지 알아보는 기술
3. 예를 들어, 얼굴이 웃고 있으면 "기쁨", 눈썹이 찌푸려 있으면 "슬픔"이나 "화남"으로 인식해요.
4. 이 기술은 로봇, 게임, 감정 AI, 수업 집중도 분석 등에 활용돼요.

# 3. 4D/3D Facial Expression Recognition

## 1. 4D/3D 얼굴 표정 인식

### 2D 얼굴 인식:

- 사진 한 장(평면)만 보고 표정 판단  
예: 웃는 사진 → "기쁨"이라고 판별

### 3D 얼굴 인식:

- 얼굴을 입체로 인식해서 볼록한 정도, 근육 움직임, 깊이감까지 파악  
예: 입꼬리 올라가는 깊이, 눈썹의 각도 등을 더 정확하게 분석

### 4D 얼굴 인식:

- 3D 얼굴이 시간에 따라 어떻게 변하는지까지 추적!  
→ 움직임 기반 감정 인식  
예: 순간 웃었다가 사라지는 미소 → 감정이 진짜인지, 억지인지 구별 가능!

## 4. EEG: Electroencephalogram

- 뇌에서 나오는 전기 신호(뇌파)\*\*를 기계로 기록하는 방법
- 머리에 작은 센서를 붙이면, 뇌가 깨어 있을 때, 졸릴 때, 집중할 때 나오는 전기 신호가 측정돼요.
- 게임할 때 집중이 잘 되는지 확인
- 수면 상태 분석
- 간질(뇌전증) 발작 여부 진단 등

## 5. EMG: Electromyography

- 근육에서 나오는 전기 신호를 측정하고 분석하는 검사
- 사람이 팔이나 다리를 움직이려고 하면, 근육 속 신경이 전기 신호를 보내는데,
- EMG는 이 전기 신호를 감지해서 근육이 잘 작동하는지, 신경에 문제가 있는지 등을 확인 줌

## 6. ML: Machine Learning

- 기계(컴퓨터)가 스스로 데이터를 보고 배우는 기술
- 사람이 직접 "이렇게 해!"라고 다 알려주는 게 아니라,  
**컴퓨터가 많은 예제를 보면서 규칙을 스스로 학습함**
- 예시
  - ✓ **고양이** 사진을 1000장 보여주면,  
→ 컴퓨터가 "아~ 이런 게 고양이구나"라고 **스스로 특징을 배움**
  - ✓ 그다음엔 처음 보는 사진도  
→ "이건 고양이다 / 아니다"라고 **판단할 수 있게 됨**

## 7. GMM: Gaussian Mixture Model (가우시안 혼합 모델)

1. GMM은 여러 개의 정규분포를 섞어서, 데이터를 더 잘 설명하는 모델
2. 통계기반 기지도 학습 모델
3. **GMM은 수학 함수 생긴 모델**  
"입력을 넣으면 "이 데이터가 어떤 그룹에 속할 가능성이 높다"는 확률 값을 보여줘요."  
→ 그래서 분류, 군집화, 이상 탐지 등에 쓰입니다!
4. **AI·데이터 분석·신호처리 등 여러 분야에서 아주 유용하게 쓰이는 도구**



## 8. NB: Naive Bayesian (나이브 베이즈)

1. 베이즈 정리를 사용하는 간단하고 빠른 확률 기반 분류 알고리즘
2. 각 특징(단어, 숫자 등)이 서로 독립이라고 가정하고, 각 특징이 특정 결과에 속할 확률을 계산해서 분류하는 모델
3. 예) 이메일에 들어 있는 단어들을 보고,  
→ 그 이메일이 **스팸인지 아닌지** 판단해야 해요.
4. 베이즈 정리: 토머스 베이즈, 영국 수학자  
기존확률+새로운 정보 반영해서 => 다시 확률 계산

**“새로운 단서를 알게 되면, 내가 생각하는 확률도 달라져야 해요!”**

## 9. LDA: Linear Discriminant Analysis (선형 판별 분석)

- 바닥에 사과랑 당근을 섞어 놓았다고 해요.
- 그런데 우리는 "이 줄을 기준으로 왼쪽은 사과, 오른쪽은 당근!" 하고 나누고 싶어요.
- LDA는 이런 '**최적의 선(또는 평면)**'을 찾아주는 **방법**이에요!
- 함수 형태로 정의되고, 그래프로 시각화 할 수 있다.

관점	표현 방식	설명
수학	$y = w^T x + b$	선형 판별 함수
시각화	직선 또는 평면	데이터를 나누는 기준선
머신러닝	분류기 모델	어떤 클래스에 속할지 결정

# 10. DT: Decision Tree (결정 트리)

- 조건을 따라가며 결과를 예측하는 분기 알고리즘
- 조건 기반의 분기 구조
  - ✓ 특정 조건을 기준으로 데이터를 여러 갈래(분기)로 나누어 처리하는 구조
  - ✓ 주어진 입력값에 따라 **"예 / 아니오"**, 또는 **"조건 만족 / 불만족"** 식으로 다음 단계로 나아가며 **결과를 도출**하는 방식
- 어디에 쓰이나?
  - ✓ 결정 트리(Decision Tree)
  - ✓ 조건문 기반 알고리즘 (if-else, switch 등)
  - ✓ **설문지, 진단 시스템** 등
  - ✓ 게임 대화 흐름, 챗봇 설계, UI 흐름 설계

# 11. KNN: K-Nearest Neighbors (K-최근접 이웃)

- K-최근접 이웃 알고리즘
- 주변 이웃들이 누구냐에 따라 결과를 정하는 알고리즘
- 예: 주변에 고양이 사진이 많으면, 이것도 고양이라고 판단
  - ✓ 먼저 훈련용 데이터가 있어요: 고양이 사진 100장 + 강아지 사진 100장(각각 어떤 동물인지 정답이 붙어있음)
  - ✓ 각 사진은 컴퓨터가 **숫자 벡터로 바꿔서 저장**해요: 예: "귀 길이", "코 위치", "털 색상" 같은 특징 → [0.2, 0.8, 0.3, ...]
  - ✓ 새로 들어온 사진을 보고 컴퓨터가 그 특징을 계산: 주변에 비슷한 사진은 뭐가 있을까? 거리 계산
  - ✓ 가장 가까운 K개의 사진을 고름: 그 중 가장 많은 사진 다수결로 결정

## 12. HMM: Hidden Markov Model (숨겨진 마르코프 모델)

- 겉으로는 안 보이는(숨겨진) 상태를, 겉으로 보이는 단서를 통해 추측하는 알고리즘

✅ 예시 1: 친구의 기분 맞추기 게임 🤖

🎯 문제:

- 당신은 친구의 기분(좋음/나쁨)을 알고 싶어요
- 그런데 친구는 기분을 말해주지 않아요 (→ ! 숨겨진 상태)

🔍 당신이 볼 수 있는 건?

- 친구가 보낸 문자 메시지  
→ "오늘 너무 좋아~", "아 피곤해..."  
(→ ✅ 겉으로 보이는 단서)

🤖 HMM은 이렇게 해요:

- 문자 메시지(단서)를 분석해서  
→ "이 메시지면 기분이 좋을 확률이 몇 %, 나쁠 확률이 몇 %"  
→ 그래서 "지금 기분은 '좋음' 상태일 확률이 높다"라고 추론

## 13. ANN: Artificial Neural Network (인공 신경망)

- 사람 뇌처럼 정보를 주고받으며 학습하는 컴퓨터 시스템

### 사람은 이렇게 생각해요:

- 눈으로 고양이 사진을 보면
- 뇌 속 신경세포(뉴런)들이 반응해서
- "어! 이건 고양이야!" 하고 판단하죠?

### ANN도 이렇게 작동해요:

1. 고양이 사진 → 숫자로 바뀜 (입력층)
2. 숫자들이 여러 뉴런을 거치며 계산됨 (은닉층)
3. 마지막에 "고양이일 확률 98%" 같은 결과 나옴 (출력층)

→ 이걸 학습시키면 점점 더 잘 맞히게 됩니다!

# 14. PCA: Principal Component Analysis (주성분 분석)

- 많이 복잡한 정보를, 중요한 것만 뽑아서 간단하게 만드는 기술
- 마치 큰 그림을 작게 요약하면서, 핵심은 놓치지 않는 방법

단계	설명
①	데이터를 좌표처럼 배치
②	가장 많이 퍼진 방향 찾기 (정보가 제일 많은 방향)
③	그 방향으로 데이터를 눌러서 투영
④	덜 중요한 축 정보는 버림
	결과: 작지만 중요한 정보만 남음!

markdown



이 점들이 나타내는 건 복잡한 정보예요.  
(예: 고양이 사진의 픽셀, 시험 점수, 센서 데이터 등)

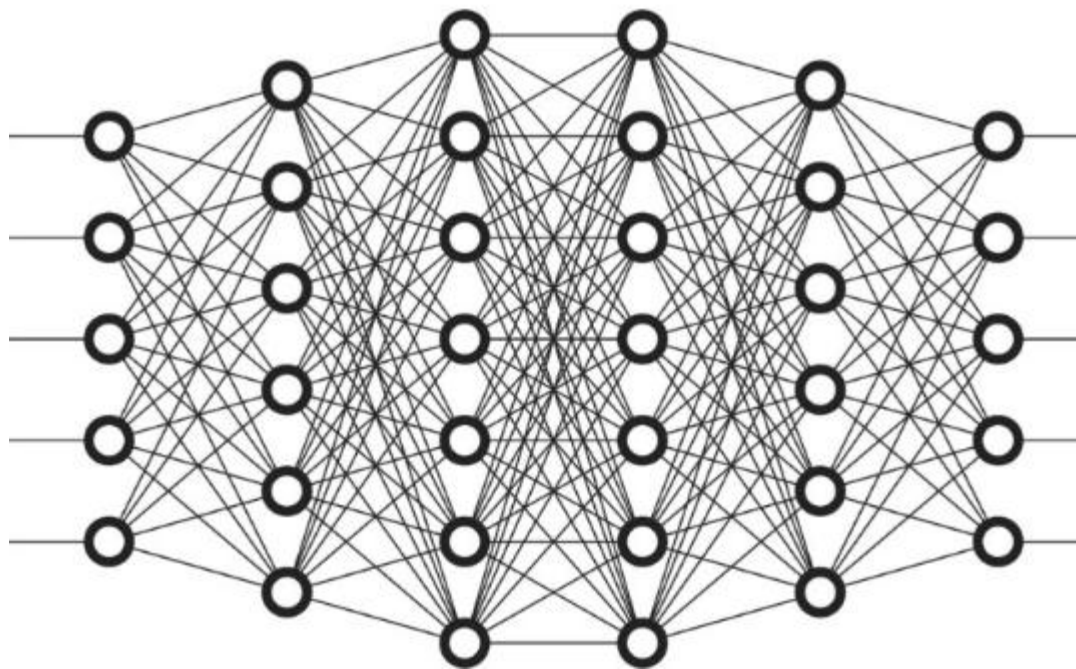


• • • • •

- 이건 더 적은 수치(예: 1차원 또는 2차원)로
- 원래 데이터의 핵심 패턴만 남긴 모습

## 15. MLP: Multi-layer Perceptron (다층 퍼셉트론)

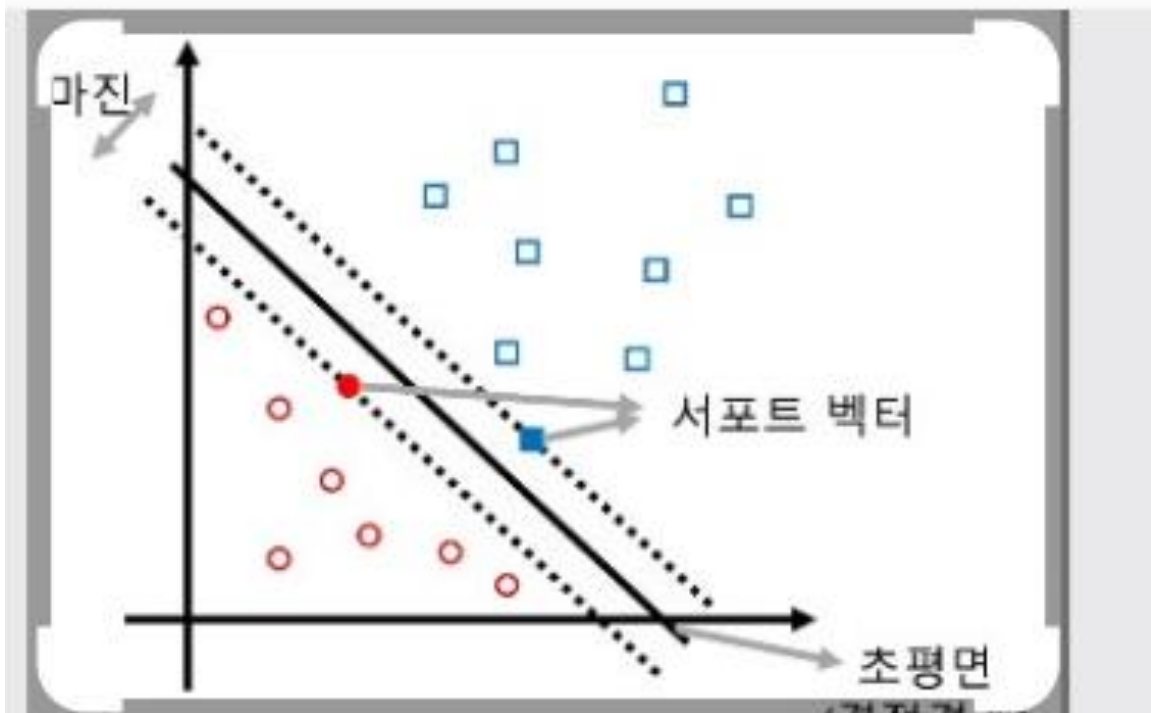
- 여러 층으로 구성된 인공 신경망
- 퍼셉트론이라는 뇌세포 비슷한 계산기들을 **층층이 쌓아** 더 복잡한 문제도 풀 수 있게 만든 구조





## 16. SVM: Support Vector Machine (서포트 벡터 머신)

- 데이터를 가장 깔끔하게 나누는 '최적의 선(또는 경계선)'을 찾아주는 분류 알고리즘



초평면	데이터를 나누는 기준선(또는 면)
마진	기준선과 가장 가까운 데이터 간의 거리 (넓을수록 안정적)
서포트 벡터	마진의 경계를 결정하는 핵심 데이터 (선의 위치를 결정함)

## 17. RBM: Restricted Boltzmann Machine (제한된 볼츠만 기계)

- 데이터를 보고 그 안의 '숨은 패턴'을 학습하는 확률 기반 인공 신경망
- **Restricted(제한된)** : 학습할 때 특정 연결 규칙을 지키도록 설계된다는 뜻

### 예 1: 층끼리 연결 제한

입력 노드끼리는 연결하지 않기 ✖  
숨겨진 노드끼리도 연결하지 않기 ✖

✓ 오직

입력층(Visible) ↔ 숨겨진층(Hidden)

→ 이렇게만 연결 가능!

(→ 이게 "제한된" 구조의 핵심 규칙이에요)

### 이유?

- 계산이 단순해지고
- 확률 계산이 독립적으로 가능해짐
- 학습이 더 빠르고 안정적

## # RBM 학습 예시

### 📌 1. 숨은 패턴 관찰 (훈련 데이터 분석)

"많은 사용자들이 ①, ②, ④, ⑤번 액션 영화를 보고 나면,  
자연스럽게 ③번 영화도 함께 보더라."

→ 이걸 직접 사람이 규칙을 만든 게 아니라,

RBM이 많은 사용자 데이터를 관찰하면서 스스로 발견한 패턴입니다.

---

### 📌 2. 확률로 예측 (새로운 사용자에게 적용)

"이 사용자(A)가 ①, ②, ④, ⑤를 봤네?  
이건 우리가 전에 자주 봤던 조합이야.  
그럼 이 사람도 ③을 좋아할 확률이 높겠네!"  
→ 그래서 확률적으로 ③번을 추천하는 것이예요.

✓ 이때 정해진 규칙 없이,

통계적으로 자주 나타난 연결을 바탕으로 확률을 추정합니다.

→ 바로 확률 기반 패턴 학습!

## 18. RBF: Radial Basis Function (반지름 기저 함수)

- Radial: 중심에서 뻗어 나가는, 반지름 반향의~
- 입력값이 중심점에서 얼마나 떨어져 있는지(반지름)에 따라 반응하는 함수
- 중심점을 기준으로 '반지름 거리만큼 떨어진 점들은 같은 값을 가진다.
- 중심에서 거리만 중요하고, 방향은 중요하지 않다.

<예시>

AI가 사진을 보면,

“이게 내가 알고 있는 고양이 특징과 얼마나 가까운가?”

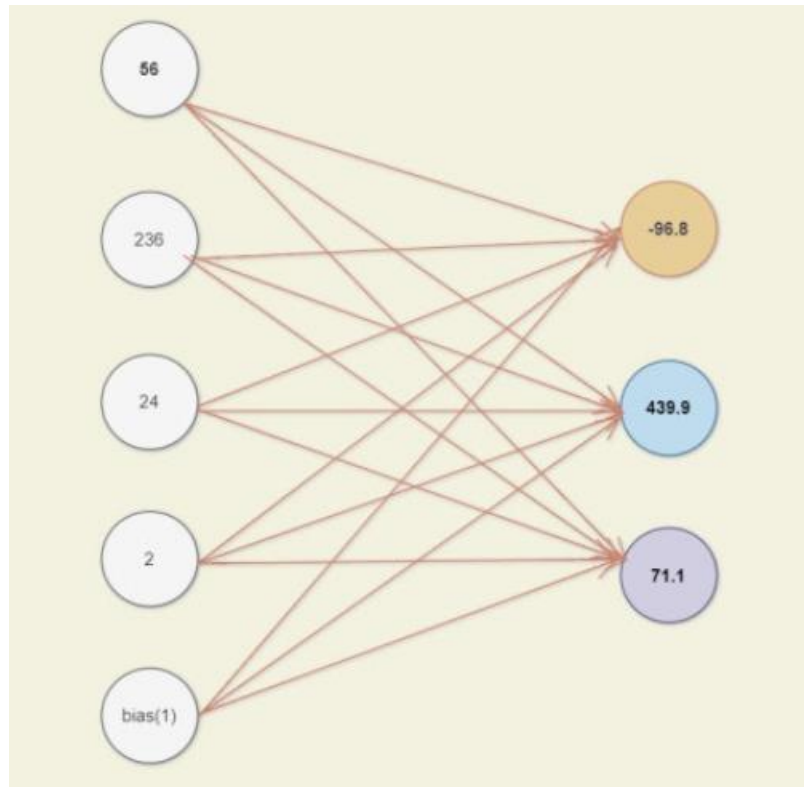
를 RBF 함수로 계산해서 판단해요.

거리 가까우면 → 반응 세다,

멀면 → 반응 약하다



# 19. FC: Full-connected (완전 연결)

- 신경망에서 한 층의 모든 뉴런이 다음 층의 모든 뉴런과 연결된 구조



## 20. MKL: Multiple Kernel Learning (다중 커널 학습)

- 여러 개의 서로 다른 커널 함수들을 동시에 사용해서, 데이터를 더 잘 분류하거나 예측하려는 기법
- 커널 함수란? 데이터를 더 잘 나눌 수 있도록 공간을 휘어서 변형시키는 함수
- Kernel = 핵심, 중심, 씨앗

 비유: 여러 개의 안경을 써보기 

- 사람 얼굴을 인식하고 싶어요.
- 선글라스를 쓰면 모양만 보이고,
- 적외선 안경을 쓰면 열 분포가 보이고,
- 고글을 쓰면 윤곽이 보일 수 있어요.

각 안경 = 하나의 커널 함수

얼굴을 제대로 알려면 하나만 쓰지 말고, 여러 안경을 같이 써보는 게 더 정확하겠죠?

 이게 바로 **Multiple Kernel Learning**이에요!

## 21. RF: Random Forest (랜덤 포레스트)

- 혼자 판단하지 말고, 숲처럼 여럿이 같이 판단하는 결정 알고리즘

표현 요소	랜덤 포레스트 개념과의 연결
“혼자 판단하지 말고”	→ 하나의 결정 트리(Decision Tree)는 예측력이 낮을 수 있음 (불안정)
“여럿이 같이 판단”	→ 여러 트리를 모아서 투표하거나 평균을 냄 (앙상블)
“숲처럼”	→ 트리(tree) 여러 개 = forest(숲)
“결정 알고리즘”	→ 각각의 트리는 조건에 따라 판단하는 결정 기반 모델

## 22. ICA: Independent Component Analysis (독립 성분 분석)

- '섞인 신호'를 보고, 그 안에 숨겨진 '독립적인 원래 신호'들을 분리해주는 알고리즘

예시

여러 사람 목소리 섞임

뇌파 데이터 (EEG)

겹친 이미지

ICA가 하는 일

→ 각 사람 음성 분리

→ 뇌 영역 신호 분리

→ 원본 이미지 분리

### 예시 1: 칵테일 파티 문제 (대표 사례)

상황:

- 당신은 파티에 있고,
- 2명이 동시에 말하고 있어요
- 마이크 2개가 소리를 섞어서 녹음했어요:



마이크 1 = A목소리 + B목소리

마이크 2 = A목소리 + 약간 다른 각도로 B목소리

! 문제: 누가 무슨 말 했는지 모르겠어요!

ICA의 역할:

ICA는 이 섞인 신호 2개를 분석해서

→ A의 목소리, B의 목소리를 따로 분리합니다!  



## 23. BoW: Bag-of-Words (단어 집합 모델)

- 문장을 구성하는 단어들의 등장빈도를 고려해서, 문장을 숫자벡터로 바꾸는 아주 단순한 텍스트 표현 방식

<두 문장> I love AI, AI loves me

단어	I	love	AI	loves	me
문장 1 벡터	1	1	1	0	0
문장 2 벡터	0	0	1	1	1

## 24. LBP-TOP: Local Binary Pattern from Three Orthogonal Planes

(세 개의 직교 평면에서의 로컬 이진 패턴)

- "영상 속 움직임"과 "표면 패턴(질감)"을 동시에 잡아내는 3차원 패턴 인식 기술

예: 얼굴 표정 변화 감지 🙄 😊 😲

- 사람의 얼굴이 시간에 따라 미세하게 움직일 때,
- 단순히 한 장의 사진(LBP)으로는 부족하죠?

✓ 그래서 XY, XT, YT 평면에서 동시에  
얼굴의 공간 + 시간 + 움직임 패턴을 추출합니다!

평면	의미
XY 평면	정적인 이미지 (공간 - 얼굴의 질감)
XT 평면	시간에 따른 X 방향 변화 (예: 좌우 눈썹 움직임)
YT 평면	시간에 따른 Y 방향 변화 (예: 입꼬리 위아래 변화)

## 25. SER: Speech Emotion Recognition (음성 감정 인식)

- 기계가 말하는 사람의 감정을 분석해내는 기술

### 🎯 예시 상황

#### 1. 콜센터 고객 상담

- 고객이 말해요:  
"진짜 너무 답답하네요..." 🙄  
→ SER은 분노 감정을 감지하고,  
→ 상담사 화면에 "주의: 고객 화남!" 표시

특징	설명
 Pitch(음높이)	화난 사람은 음이 높아지고 빠름
 Energy(세기)	슬플 땐 조용하고 단조로움
 Speed(말속도)	긴장/흥분 시 말이 빨라짐
 Mel-frequency features (MFCC)	사람 귀처럼 소리의 주파수 특성 분석

✓ 이런 특징들을 딥러닝, SVM, CNN, RNN 같은 모델에 넣어서 감정 분류!

## 26. FMER: Facial Micro-Expression Recognition (얼굴 미세 표정 인식)

- 얼굴에서 아주 짧고 미세하게 나타나는 표정을 감지해서, 숨겨진 감정이나 심리를 분석하는 고급 감정 인식 기술

항목	일반 표정 인식	미세 표정 인식 (FMER)
표정 시간	1~2초 이상	0.1~0.5초 이하
감정 종류	의식적 감정	무의식적 진짜 감정
난이도	비교적 쉬움	매우 어려움 (섬세)
응용	감정 분석, 게임	심리 분석, 수사, 정신 건강, 거짓말 탐지 등

## 27. EBGR: Emotional Body Gesture Recognition (감정 신체 동작 인식)

- 사람의 몸짓, 자세, 움직임만 보고, 그 사람이 어떤 감정을 느끼고 있는지를 AI가 알아내는 기술

### 예시 1: 축구 경기 중

- 어떤 선수가 두 팔을 번쩍 들고 점프해요 🙌  
→ “기뻐요!”
- 다른 선수는 고개를 숙이고 주저앉아요 😞  
→ “슬퍼요”

➡ 얼굴 표정 없이도, 몸짓만 보고 감정이 느껴지죠?

→ EBGR은 바로 이런 걸 AI가 인식하게 해주는 기술이에요!

## 28. ECG: Electrocardiography (심전도)

- 심장 박동에서 발생하는 전기 신호를 기록하는 방법
- 우리 심장은 뛰는 순간마다 전기 신호가 생기는데,
- 그걸 피부에 붙인 전극으로 감지해서
- 그래프로 그려주는 게 ECG

항목	설명
 심박수	분당 심장 박동 수
 부정맥	심장이 불규칙하게 뛰는지
 전도 이상	신호 전달이 느려졌는지
 심장 질환	협심증, 심근경색 조기 징후

## 29. EDA: Electro-Dermal Activity (피부 전도 활동)

- 피부의 전기 전도도 변화로 감정 상태를 측정하는 방법


### 예시 상황

#### 1. 공포 영화 시청 중

- 갑자기 무서운 장면이 나옴 → 놀람 + 긴장
- 순간적으로 손에 땀이 살짝
- EDA 센서가 이를 감지 → "스트레스/흥분 상태"

# 30. DL: Deep Learning (심층 학습)


- 사람의 뇌처럼 생긴 인공 신경망을 여러 층으로 깊게 쌓아서, 데이터를 스스로 학습하고 이해하게 만드는 인공지능 기술

 예시: 고양이 vs 강아지 이미지 분류

전통 방식 (기계학습):

- 사람이 직접 "귀 모양, 눈 위치" 같은 특징을 정해야 함  
→ 사람이 똑똑해야 모델이 잘 작동

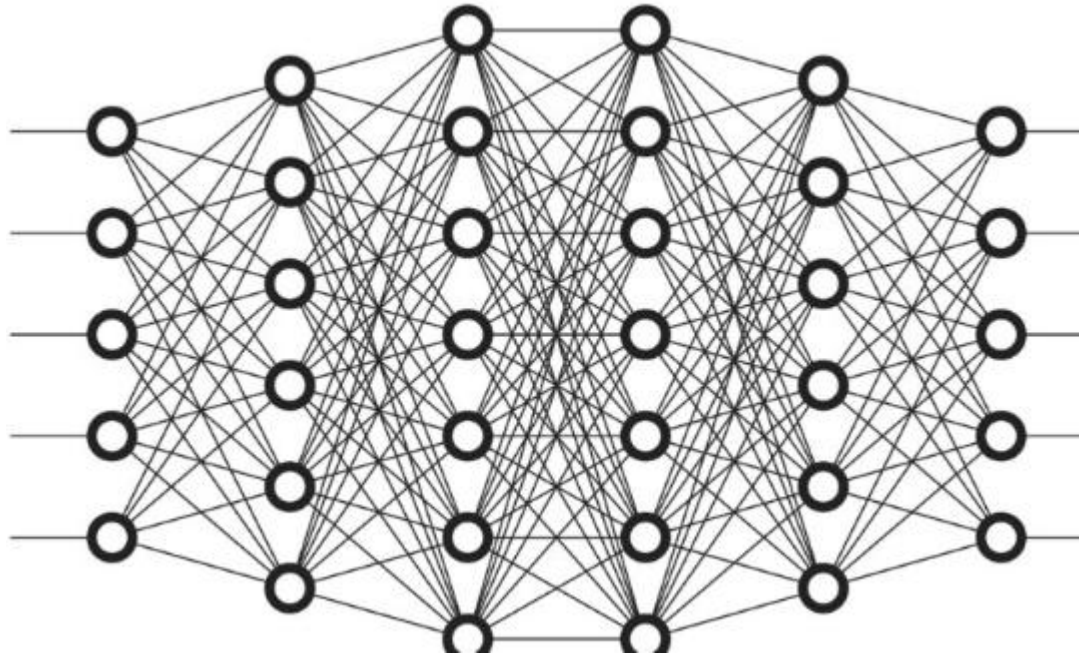
딥러닝 방식:

- 수많은 고양이/강아지 사진을 넣으면
- 딥러닝이 스스로 **\*\*"이게 귀다", "이게 코다"\*\***를 학습  
→ 사람이 특징을 안 알려줘도 됨! 



## 31. MLP: Multi-Layer Perceptron (다층 퍼셉트론)

- 여러 층으로 구성된 인공 신경망(Artificial Neural Network)



## 32. LSTM: Long-Short Term Memory (장단기 기억)


- 시간의 흐름이 있는 데이터(시계열)를 잘 이해하고 기억하는 인공신경망 모델
- 과거 정보는 기억하고, 필요 없는 건 잊고~
- 지금 중요한 정보에 집중할 수 있는 구조

# 예) 나는 너무 피곤해서 일찍 자고 싶어

- RNN

- ✓기억이 짧음 + 문장 끝에 가면 앞부분 **\*\*"피곤해서"\*\***를 잊어버려요
- ✓그래서 그냥 "자고 싶어"만 보고, "기분이 좋은가?" 착각할 수 있어요

- LSTM (기억력 좋은 AI)

- 앞에서 '피곤하다'는 말을 기억해요
- 그래서 나중에 "자고 싶다"가 나오면  
→ "아, 피곤하니까 자고 싶구나!" 하고 제대로 이해해요 

### 33. **DCNN**: Deep Convolutional Neural Network (심층 합성곱 신경망)

- 이미지를 보고 자동으로 특징을 뽑아내고, 분류까지 해주는 딥러닝 모델

사람처럼 이미지를 보고, 눈·코·귀 같은 중요한 부분을 스스로 찾아내는 똑똑한 인공지능이에요!

- ✓ CNN이 "눈, 코, 패턴"을 찾아내는 시각 전문가라면,
- ✓ DCNN은 그걸 깊고 정교하게 하는 전문가 팀!

## 예시: 고양이 사진 분류하기

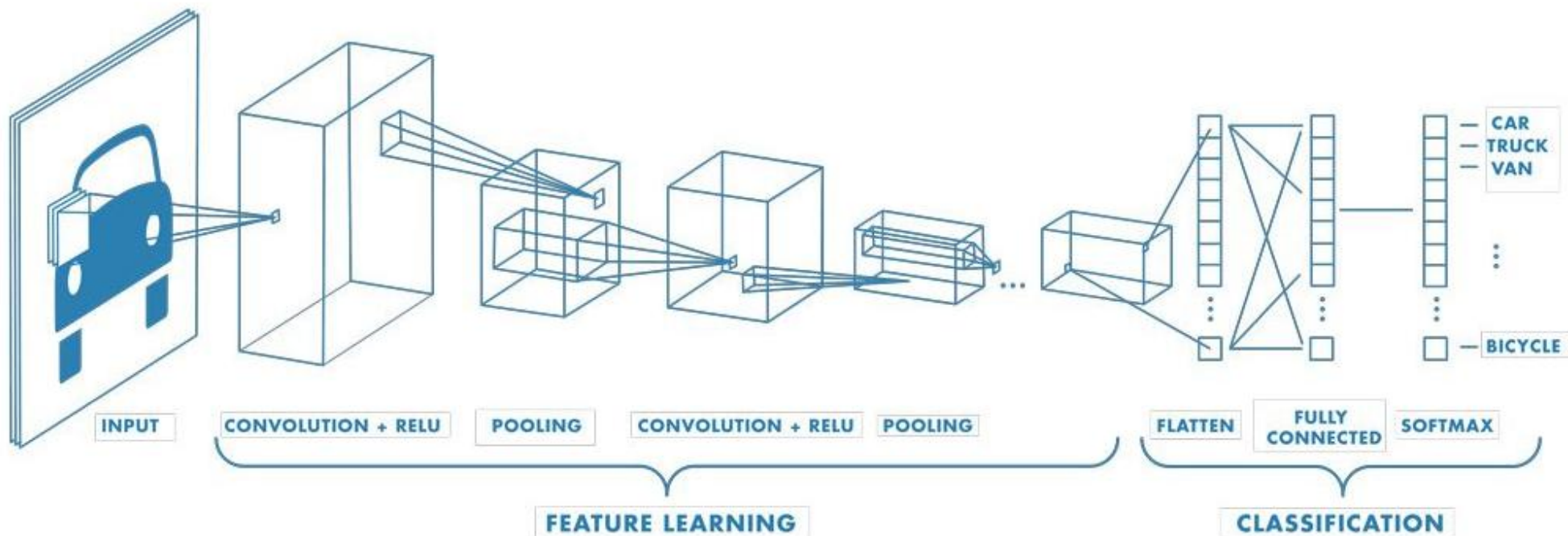
1. 사진을 넣어요
2. 초기 층: 가장자리(선, 점, 모서리)를 감지
3. 중간 층: 귀, 눈, 코, 털 패턴 같은 복잡한 특징 추출
4. 뒤쪽 층: 이건 고양이다! 라고 결정(분류)

## ✓ DCNN의 핵심 구조

구성 요소	역할
 <b>Convolution Layer (합성곱 층)</b>	이미지에서 특징 추출 (필터처럼 작동)
 <b>Pooling Layer (풀링 층)</b>	중요한 정보만 남기고 압축
 <b>Fully Connected Layer</b>	결과를 정리하고 분류 (예: 고양이 vs 강아지)
 여러 층으로 쌓여서 "Deep" 구조	단순 → 복잡 패턴으로 점점 학습

## 34. CNN: Convolutional Neural Network

- 이미지를 잘 이해하도록 설계된 딥러닝 구조
- 사진·영상 속의 특징(눈, 코, 경계, 패턴 등)을 자동으로 뽑아서 분류





## CNN vs DCNN

항목	CNN	DCNN
층 수	적음 (얕음)	많음 (깊음 = Deep)
용도	간단한 이미지 분류	복잡한 시각 인식
계산량	상대적으로 작음	훨씬 많음 (GPU 필요)



## 35. RNN: Recurrent Neural Network (순환 신경망)

- 시간 순서가 있는 데이터를 처리하기 위해 고안된 신경망
- 과거의 정보를 기억하며, 다음에 나올 것을 예측하거나 이해할 수 있음

문장:

“나는 피자를 먹고 싶어.”

- 일반 신경망: 각각 단어를 독립적으로 봄 → 연결 이해 어려움 ❌
- RNN: “나는 → 피자를 → 먹고...” 를 순서대로 기억하며  
→ “싶어”가 나올 거라고 예상! ✅

→ 문장의 문맥과 흐름을 이해할 수 있어요.


“나는 → 피자를 → 먹고 → 싶어”

- 일반 신경망: 각 단어만 보고 대답  
→ 기억이 없음 ✕

● RNN 학생:

1. “나는” → 기억함
2. “피자를” → “나는”과 연결해서 이해
3. “먹고” → “나는 피자를”을 생각하며 해석
4. “싶어” → 전체 문장을 기억하며 처리함

## 예시: 손글씨 숫자 인식

- 예: 숫자 3을 CNN에 넣어요
  - 1. **첫 번째 층:** 선, 곡선 같은 단순한 모양 찾기
  - 2. **중간 층:** 숫자 전체 윤곽, 특징 감지
  - 3. **마지막 층:** "이건 숫자 3이야!" 라고 판단! 
- 
- ➡ 사람이 "이게 눈이다", "이건 털이다"라고 안 알려줘도
  - ➡ CNN이 **자동으로 이미지 특징을 학습**해요!

## 36. **GRU: Gated Recurrent Unit** (게이트 순환 유닛)

1. RNN(Recurrent Neural Network)\*\*의 한 종류
2. 기본 RNN은 기억력이 너무 짧다. 그래서 문제를 해결하기 위해 나온 게 **LSTM**과 **GRU**이다.
3. 2개의 게이트를 사용함
  - Update Gate: 과거 정보를 얼마나 유지할지 결정
  - Reset Gate: 과거 정보를 얼마나 잊을지 결정
  - 두 게이트 협력해서 필요한 정보만 기억하고, 불필요한 정보는 날려버림

## **VS** GRU vs LSTM vs RNN 정리

모델	구조	기억 능력	학습 속도	사용 예
RNN	간단	낮음	빠름	간단한 시계열
GRU	중간 (2개 게이트)	높음	빠름	대다수 NLP, 시계열
LSTM	복잡 (3개 게이트)	매우 높음	느림	긴 문장, 복잡한 구조

구분	의미	예시
HW 게이트	논리 회로	AND, OR, NOT
GRU 게이트 (SW)	수학 연산으로 정보를 조절	시그모이드, 행렬 연산 등

- GRU에서의 게이트는 "소프트웨어적 개념"이며, 뉴런의 활성화 함수와 가중치를 이용해 계산되는 수학적 장치

## 37. AE: Auto-encoder (오토인코더)

1. 일반적인 인코더/디코더 시스템: 인코더(입력을 압축), 디코더(그 압축된 벡터를 다른 데이터로 바꿔줌)
  - 예) 기계번역에서 영어 → 인코더 → 벡터 → 디코더 → 한국어
  - 입력과 출력이 다른 종류
2. **Autoencoder는 입력과 출력이 똑같음**
  - 입력: 어떤 데이터 (예: 이미지)
  - 출력: **입력과 최대한 똑같은** 재구성 데이터
  - 즉, 자기 자신을 인코딩하고 다시 복원

“자기 자신을 복원하는 뉴런 구조라는 점에서 “자동(auto) 자기 부호화(encoding)”라는 뜻으로 붙은 이름”

## 38. GAN: Generative Adversarial Network (생성적 적대 신경망)

1. 2014년, **Ian Goodfellow**가 제안한 혁신적인 딥러닝 모델
2. 목표: 진짜 같은 새로운 데이터를 생성하는 것
3. “사기꾼과 경찰이 서로 경쟁하며 발전하는 구조”
  - 이 경쟁 구조 덕분에 매우 사실적인 이미지를 생성할 수 있음
4. 예시) 완전히 가짜인데 진짜 사람처럼 보이는 얼굴
  - <https://thispersondoesnotexist.com/> : GAN이 만든 가짜 얼굴 보여줌
  - NVIDIA의 StyleGAN 모델은 GAN의 대표적 응용
5. 기타
  - Deepfake, CycleGAN 등



## 39. VGG: Visual Geometry Group

(비주얼 기하학 그룹)

1. 이미지 분류를 위한 CNN(합성곱 신경망) 아키텍처
2. **Oxford 대학교의 Visual Geometry Group**에서 2014년 발표
  - 명칭: 연구 그룹 이름을 따서 붙임
3. 초기 CNN(AlexNet, LeNet) 단점
  - 계산량이 많음, 파라미터 많음, 일반화 부족
  - 구조 복잡, 네트워크 깊이 얇음
4. VGG 개선사항: "아이디어: 3X3 필터만 반복해서 깊게 쌓자."
  - 계산의 효율성 증가
  - 성능개선

항목	초기 CNN	VGG
필터 크기	큼 (5x5, 11x11)	작음 (3x3 반복)
구조	복잡	단순하고 반복적
깊이	얕음	깊음 (16~19층)
효율성	비효율적, 파라미터 큼	계산 효율 + 표현력 좋음
예시	큰 붓으로 대충 그림	작은 붓으로 디테일하게 그림

## 40. DBN: Deep Belief Network

(심층 신뢰 네트워크)

1. Transformer나 CNN이 나오기 전, 딥러닝 시대를 여는 데 기여한 선구자 같은 모델
2. DBN은 여러 개의 Restricted Boltzmann Machine (RBM)을 층층이 쌓아 만든 딥러닝 모델
3. 전통 딥러닝 모델의 단점(2006년 이전)
  - 기울기 소실 문제: 깊은층으로 갈 수록 기울기가 너무 작아 학습이 안 됨
  - 과적합 + 연산 자원 부족: 데이터는 적고 GPU 성능 낮음

# # DBN: Deep Belief Network

1. 2006년 **Geoffrey Hinton** (제프리 힌튼) 박사
2. 논문제목: A Fast Learning Algorithm for Deep Belief Nets
3. 발표: 2006년, *Neural Computation* 저널
4. 저자: Geoffrey Hinton, Simon Osindero, Yee-Whye The
5. 기여도
  - “**딥한 구조도 학습할 수 있다!**”는 가능성을 처음으로 보여줌
  - 신경망이 “**얕아야 잘 동작한다**”는 기존 인식을 깨뜨림
  - 이후 딥러닝 연구 붐을 일으킴 → CNN, RNN, Autoencoder, GAN, Transformer의 시대가 열림

## # 딥러닝은 학습이 안 됐어요 (2006년 이전)

- 3~4개 층만 쌓아도 학습이 전혀 안 됨
- 기울기 소실 문제(Gradient Vanishing) 때문에 아래층까지 정보가 전달되지 않음
- 가중치 초기값이 나쁘면 망한 학습에서 회복이 불가능
- GPU도 없던 시절이라 학습 시간도 오래 걸림

## # DBN의 혁신적 해결책: 사전학습 (Pretraining)

- "한 번에 전체를 학습하지 말고, \*\*한 층씩 차근차근 배우게 하자!"
- \*\*RBM(Rrestricted Boltzmann Machine)\*\*을 한 층씩 쌓고,
- 각 층은 **자기 바로 아래 층의 출력을 입력으로 사용**해서 비지도 학습을 진행함.
- 각 RBM은 입력 데이터를 받아 **고수준 특징**을 추출: 이걸 다시 다음 층의 입력으로 사용
- 모든 층을 사전학습(pretraining)한 후, 마지막에 **전체 네트워크를 fine-tuning** (지도학습)으로 조정

## 쉽게 비유하면...

### 전통 신경망 (깊은 네트워크):

“10층 건물을 설계도 없이 한 번에 짓는다”  
→ 기초부터 흔들리고, 결국 무너짐

### DBN:

“각 층을 차례로 튼튼하게 짓고, 마지막에 전체 마감한다”  
→ 안정적이고 높은 건물 완성 가능!

## "딥한 구조도 학습할 수 있다"는 뜻은?

과거에는:

- "깊은 모델 = 학습 안 됨"이 통설
- CNN도 2~3층 이상은 거의 안 썼음

DBN 이후에는:

- 5층, 10층짜리 모델도 학습 가능!
- 그 덕분에 ResNet(100+층), BERT(100M+ 파라미터) 같은 초대형 모델도 등장 가능



# 41. HAN: Hierarchical Attention Network (계층적 주의 네트워크)

1. 문서 분류(Document Classification)\*\*를 위해 제안된 딥러닝 모델
  - 문서를 보고, 그 문서가 어떤 "범주(클래스)"에 속하는지 자동으로 분류하는 것
  - 뉴스 기사 분류, 감정분석, 이메일 스팸 분류
2. 제안 논문: **Hierarchical Attention Networks for Document Classification**« Yang et al., 2016 (ACL)
3. 전통적인 방법: BoW + 머신러닝
  - BoW(Bag-of-Words): 단어의 순서를 무시하고 등장 횟수만 보는 방식
  - 단점: 단어 순서와 문맥 무시, 의미 파악 부족

# # HAN (Hierarchical Attention Network)

## 1. 문장을 단어들의 시퀀스, 문서는 문장들의 시퀀스로 보는 계층적 구조 모델

- 단어 수준 → 문장 수준 → 문서 수준으로 특징 추출
- 주의(attention)로 중요한 단어/문장에 더 집중

“하나의 긴 글(문서)을 보고, 이 글이 무슨 내용인지 알아내는 똑똑한 인공지능”

“단어 하나하나를 보고 → 중요한 문장을 찾고 → 그 문장들을 보고 전체 내용을 파악하자!”

## 42. ResNet: Residual Networks (잔차 네트워크)

- 깊은 신경망도 쉽게 학습할 수 있게 만든 CNN 아키텍처
- 아이디어: **“어려우면 그냥 원래 입력이라도 지나가게 하자!”**

예: 이미지에서 ‘고양이’ 특징 뽑기

- 어떤 층에서 입력 이미지를 받았는데,  
이미 고양이임이 충분히 드러나 있어요.
- 기존 CNN: “이걸 또 다시 가공하려 함” → 불필요하거나 오히려 정보 손실
- ResNet:
  - “지금 정보 괜찮아, 그냥 넘기자” (→ skip connection)
  - “조금만 더 털 무늬 정보 추가하자” (→ residual 학습)

## 43. GAP: Global Average Pooling(전역 평균 풀링)

“CNN 안에서 사용되는 ‘연산 방법 중 하나’

<b>Conv</b> (합성곱)	사진을 보고 <b>중요한 부분 찾기</b>	“고양이의 귀는 어디 있지?” 같은 걸 똑똑하게 찾아냄
<b>Pooling</b> (풀링)	중요한 부분만 <b>썩 썩 요약하기</b>	“눈, 코, 입 다 있네! 그럼 이걸 짧게 요약해보자!”
<b>GAP</b> (Global Average Pooling)	그림 전체를 <b>한 줄 요약하기</b>	“전체 특징을 숫자 하나씩 평균 내서 정리하자!”
<b>Softmax</b> (소프트맥스)	누구인지 <b>결정하기</b>	“고양이일 확률 80%, 강아지 20% → 고양이로 하자!”

## 예를 들어, 고양이 사진을 넣으면...

### 1. Conv

→ "이건 귀고, 이건 수염이네!" → 특징 뽑기

### 2. Pooling

→ "귀, 눈, 수염 중 중요한 것만 요약!"

### 3. GAP

→ "전체 특징을 평균 내서 한 줄로 만들자!"

### 4. Softmax

→ "이건 고양이일 확률이 더 높네!" → 고양이로 분류

## 44. AUs: Action Units (동작 단위)

1. 표정 분석을 위해 사용되며, FACS (Facial Action Coding System)라는 체계 속에 있음

AU 번호	동작 단위 설명	예시 표정
AU1	눈썹 안쪽 올리기	놀람, 집중
AU2	눈썹 바깥쪽 올리기	놀람, 호기심
AU6	볼 들어올리기 (눈가 주름)	진짜 웃음 (Duchenne Smile)
AU12	입꼬리 올리기	미소 😊
AU14	입꼬리 옆으로 당기기	경멸 😏

## 45. AAM: Active Appearance Model (능동 외모 모델)

1. AAM은 얼굴(또는 외모)의 "형태(Shape)"와 "표면 정보 (Appearance)"를 함께 정의하고 설명하는 모델
2. 외모를 수치화해서 설명하는 모델
3. 정의하는 대상: 눈, 코, 입의 위치(모양) + 피부 색상, 주름, 그림자 등(텍스처)
4. 어디에 사용하는가?
  - 얼굴 인식, 표정 분석, 얼굴 정렬, 외모 비교, 시뮬레이션 등

# Shape (모양) 예시: 얼굴의 형태를 **\*\*점(point)\*\***으로 정의

부위	좌표 예시 (x, y)	의미
왼쪽 눈끝	(34, 58)	왼쪽 눈의 바깥 부분
오른쪽 눈끝	(78, 59)	오른쪽 눈 바깥
코끝	(56, 82)	코의 중심
입꼬리 왼쪽	(42, 102)	웃을 때 움직이는 위치



# Appearance (텍스처) 예시

위치	텍스처 정보 (R, G, B or 밝기값)	의미
왼쪽 눈 아래	(120, 115, 110) → 어두운 그림자	다크서클 있음
볼	(210, 190, 180) → 밝은 톤	피부 밝음
입술	(180, 60, 80) → 붉은색	립스틱 or 혈색
이마	(200, 200, 180)	고른 피부 톤

## 46. LFPC: Logarithmic Frequency Power Coefficient (로그 주파수 파워 계수)

1. 음성 처리나 음향 신호 분석 개념
2. 사람 목소리를 숫자로 요약해서 컴퓨터가 이해할 수 있도록 만드는 과정(로그 주파수로 표현)
3. 왜 로그 주파수로 표현하는가? 사람이 비율에 민감하기 때문
  - 스피커 볼륨을 1에서 2로 올리면 → "엄청 커졌네!"
  - 9에서 10으로 올리면 → "거의 차이 없네..."

### #로그값 사용

- 큰 수를 작은 수로 표현함
- 계산이 쉽다
- 다루기 쉽다

주파수 (Hz)	로그값 $\log_{10}(\text{frequency})$	로그 차이
100	$\log_{10}(100) = 2.0$	—
200	$\log_{10}(200) \approx 2.30$	+0.30
300	$\log_{10}(300) \approx 2.48$	+0.18
400	$\log_{10}(400) \approx 2.60$	+0.12
500	$\log_{10}(500) \approx 2.70$	+0.10
1000	$\log_{10}(1000) = 3.0$	+0.30 (전체적으로 보면 작음)

## 47. ROIs: Regions of Interest (관심 영역)

“이미지나 영상에서 ”특히 중요해서 집중하고 싶은 부분“

### 예시 1: 얼굴 인식

- 사진 전체가 입력인데
- AI는 그중에서 **눈, 코, 입, 얼굴 영역만 집중**해서 보고 싶어함  
➡ 이게 바로 ROI (관심 영역)!

### 예시 2: 의료 영상 (CT, MRI 등)

- CT 영상 전체가 1000×1000이라도
- 의사는 **종양이 있는 작은 영역**만 보고 분석하고 싶음  
➡ 이 작은 부분이 ROI!