Pooling in Convolutional Neural Networks

A comprehensive guide to the relationship between **stride (S)** and **window size (F)** in both **Max** and **Average** pooling, their definitions, effects on the network, use-cases, and pitfalls of choosing S too low or too high.

---

1. Definitions

- **Pooling Window (F)**
  The spatial footprint (height × width) over which we aggregate activations.
  - Common choices: 2×2, 3×3, 5×5.
- **Stride (S)**
  The step size by which the pooling window moves across the feature map.
  - When $S = F$, windows tile exactly.
  - When $S < F$, windows **overlap**.
  - When $S > F$, windows leave **gaps**.
- **Max Pooling**

$$y_{i,j} = \max_{(u,v) \in \mathcal{W}_{i,j}} x_{u,v}$$

  Picks the maximum activation in each window $\mathcal{W}_{i,j}$.
- **Average Pooling**

$$y_{i,j} = \frac{1}{|\mathcal{W}_{i,j}|} \sum_{(u,v) \in \mathcal{W}_{i,j}} x_{u,v}$$

  Computes the mean of all activations in each window.

---

2. Relationship Between S and F

2.1. Exact Tiling: $S = F$

- **Coverage:** Each pixel belongs to exactly one window.
- **Output size:**

$$H_{\text{out}} = \left\lfloor \frac{H_{\text{in}} - F}{F} \right\rfloor + 1, \quad W_{\text{out}} = \left\lfloor \frac{W_{\text{in}} - F}{F} \right\rfloor + 1$$

- **Typical choice:** Ensures predictable downsampling by factor F.

2.2. Overlapping Windows: $S < F$

- Windows share $F - S$ rows/columns with neighbors.
- **Output size shrinks slowly:**

$$H_{\text{out}} = \frac{H_{\text{in}} - F}{S} + 1$$

- **Effect:**
  - **Smoother features** (redundant coverage).
  - **High compute & memory** (many overlapping windows).

2.3. Gaps Between Windows: $S > F$

- Windows jump past $S - F$ rows/columns, leaving blind spots.
- **Output size shrinks faster:**

$$H_{\text{out}} = \left\lfloor \frac{H_{\text{in}} - F}{S} \right\rfloor + 1$$

- **Effect:**
  - **Information loss** (pixels never pooled).

- **Aliasing artifacts** (under-sampling).

---

3. Effects on Max vs. Average Pooling

Aspect	Max Pooling	Average Pooling
Overlapping ($S < F$)	Highly redundant maxima; very smooth activations	Extra low-pass blur; over-smoothed maps
Tiling ($S = F$)	One strongest activation per block	Clean, anti-aliased downsampling
Gaps ($S > F$)	May drop “hot” pixels entirely → sparse output	Biased means; severe aliasing
Translation Invariance	Good when tiled; redundant when overlapped	Good smoothing when overlapped; risk of bias

---

4. Use-Cases & Recommendations

Scenario	Recommended Pooling	Notes
Standard image classification	2×2 window, $S = 2$ (Max-pool)	Efficient down-sampling, preserves strong features
Anti-checkerboard in generators	3×3 window, $S = 2$ (Avg- or Max-pool)	Mild overlap for smoother feature maps
Very deep nets needing context	3×3 window, $S = 1$ (Avg-pool)	Overlap slows downsampling, grows receptive field gradually
Extreme spatial reduction	2×2 window, $S = 4$ (Max- or Avg-pool)	Aggressive tiling with gaps, but risks lost info

---

5. Pitfalls of Low Stride ($S < F$)

1. **Compute & Memory Blow-Up**
   - ≈ $H \cdot W$ windows per layer instead of ≈ $(H/S) \cdot (W/S)$.
2. **Slow Spatial Reduction**
   - Feature maps remain large → slower deeper layers.
3. **Gradient Congestion**
   - Many overlapping windows back-propagate through the same pixels → unstable updates.
4. **Border & Padding Issues**
   - Asymmetric truncation at edges if not padded perfectly.

---

6. Pitfalls of High Stride ($S > F$)

1. **Information Loss**
   - Pixels in gaps are never pooled — risks missing critical features.
2. **Aliasing & Checkerboarding**
   - Under-sampling can introduce artifacts, especially in generative contexts.
3. **Uncontrolled Receptive-Field Jumps**
   - Neurons skip context; might miss mid-scale patterns.

---

7. Impact on Model Behavior

- **Receptive Field Growth:**
  - **Slow** when $S < F$; **fast** (but coarse) when $S > F$.
- **Translation Invariance:**
  - **Best** at $S = F$; **redundant** when overlapped; **poor** when gapped.

- **Resource Utilization:**
  - **CPU/GPU:** More tasks with $S < F$; fewer—but riskier—tasks with $S > F$.
- **Training Stability:**
  - Overlap can "over-smooth" gradients; gaps can omit gradients altogether.

---

8. Conclusion

For **most CNN classification** tasks, the rule of thumb is:

> **Stride = Window size** $(S = F)$
> → perfect tiling, full coverage, predictable down-sampling.

Use $S < F$ (overlap) or $S > F$ (gaps) **only when** you have a **specific architectural goal** (e.g., smoothing in generative nets, extreme down-sampling, or gradual receptive-field growth). Outside those niche cases, sticking with $S = F$ minimizes computational waste, memory bloat, and ensures consistent feature-map coverage.