ASPPNeXt Architechture

# A Report

## Data Preprocessing:

* The following is kept in mind while performing augmentations on the data such as Adding shadows, glare, dust, or raindrops using augmentation libraries like albumentations to reflect field conditions which makes the architecture robust to
* Some Rules are followed to keep the Augmentation Pipeline optimized:
  1. Crop Early - RandomCrop/RandomResizedCrop first in pipeline (16x speedup possible!)
  2. Fix OpenCV Threading - Add cv2.setNumThreads(0) in PyTorch DataLoader workers
  3. Use uint8 Images - Keep images as uint8 until final Normalize
  4. Combine Transforms - Use Affine instead of separate rotate+scale
  5. Optimize Image Reading - Use OpenCV (cv2.imread) or torchvision (torchvision.io.decode\_image) instead of PIL/Pillow